

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
ECOLE NATIONALE POLYTECHNIQUE.



DEPARTEMENT D'ELECTRONIQUE

Projet de fin d'études

**En vue de l'obtention du
Diplôme d'ingénieur d'état en Electronique**

**Système de traitement d'image sur SoC
(Application biométrique)**

Proposé et dirigé par :

M. R.SADOUN

Réalisé par:

Toufik.OULED-CHEIKH
Abdelghani. LAMRI

Promotion 2008

Remerciements

Nos remerciements vont exceptionnellement à Monsieur R.SADOUN, chargé de cours à l'Ecole Nationale Polytechnique d'Alger, pour son aide, son suivi, ses conseils et directives et pour son dévouement.

Nous tenons à remercier Melle M. GUERTI, Professeur à l'Ecole Nationale Polytechnique d'Alger, d'avoir accepté de présider le jury.

Nos remerciements vont aussi à Mme L.HAMAMI, Maître de conférence à l'Ecole Nationale Polytechnique, d'avoir bien voulu examiner notre travail.

Nous remercions tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, spécialement ceux des départements des Sciences Fondamentales et d'Electronique, pour leur apport en savoir.

Nos remerciements, vont au personnel de l'Ecole et à toute personne dévouée au service de l'Ecole Nationale Polytechnique.

Enfin, nos remerciements vont à toute personne ayant contribué, de près ou de loin, à réaliser ce travail.

Dédicaces

Je dédie mon travail à mes chers parents, à mes deux grandes mères, à mon petit frère Walid et ma chère sœur Sofia qui m'a soutenu sans relâche, dans toutes les circonstances, tout au long de mon parcours d'études.

Je dédie, aussi, ce travail à toute ma famille pour leur soutien.

Je dédie ce travail à mes deux amis, mon binôme Toufik, Amine et Khalil

Je dédie ce travail à tous mes amis de Bouraoui pour leur soutien tout au long de mon parcours.

Enfin je dédie ce travail à ma chère et très spéciale promotion.

Abdelghani

*Je dédie ce travail à
Mes parents qui m'ont toujours soutenu,
Mon grand frère Amin,
Ma petite sœur Djazia,
Toute ma famille,
Tous mes amis,
Mon binôme Ghanou,
Ma chère et très spéciale promotion 2008*

Toufik

ملخص:

هذا المشروع يشتمل على تنفيذ حل التعرف على بصمات الأصابع على نظام على رقاقة (SoC). بعض الخوارزميات جرى إثبات صحتها في Matlab. بعد ذلك، تمت دراسة نهج للتنفيذ على SoC. أخيراً، الطريقة المختارة هي إدماج المكتبة FVS لأنها تتميز بخصوصيات التي تجعل هذا الحل مناسب للتنفيذ على SoC.
الكلمات المفاتيح : إحصائيات بيولوجية- بصمات الأصابع- معالجة الصور- SoC- FPGA

Résumé :

Ce projet consiste en l'implémentation d'une solution de reconnaissance d'empreinte digitale sur un système sur puce (SoC). Des algorithmes ont été validés sous MatLab. Ensuite, quelques approches ont été étudiées pour les implémenter sur SoC. Finalement, la démarche adoptée a été d'intégrer la bibliothèque FVS de part ses caractéristiques qui font d'elle, une solution adaptée pour les systèmes sur puces.

Mots clés : Biométrie- Empreinte digitale- traitement d'image- SoC- FPGA

Abstract:

This project consists in the implementation of fingerprint recognition solution on a system on chip (SoC). Algorithms were validated under MatLab. Then, some approaches were studied to implement them on SoC. The adopted method was to integrate FVS library which characteristics make it adapted for systems on chip

Keywords: Biometrics- Fingerprint- Image processing- SoC- FPGA

Table des matières

Introduction Générale	1
Chapitre I : La biométrie	
I-1 Introduction.....	3
I-2 Termes utilisés en biométrie.....	4
I-3 Les caractéristiques biométriques.....	6
I-4 Evaluation d'un système biométrique.....	11
I-5 Conclusion.....	12
Chapitre II : Traitement de l'empreinte digitale	
II-1 Définition.....	13
II-2 Caractéristiques des empreintes digitales.....	13
II-2-1 Les points singuliers globaux.....	14
II-2-2 les points singuliers locaux.....	15
II-3 Système de reconnaissance d'empreintes.....	15
II-3-1 Acquisition de l'image.....	16
II-3-2 Module d'extraction des minuties.....	17
II-3-2-1 Prétraitement de l'empreinte.....	18
II-3-2-1-1 Rehaussement d'image d'empreinte digitale.....	18
a- Egalisation d'histogramme.....	19
b- Approche fréquentielle auto adaptive.....	19
II-3-2-1-2 Binarisation de l'empreinte digitale.....	22
II-3-2-1-3 Segmentation de l'image.....	22
II-3-2-2 Extraction des minuties.....	23
II-3-2-2-1 La squelettisation de l'image.....	23
II-3-2-2-2 La détection des minuties.....	25
II-3-2-2-3 L'élimination des fausses minuties.....	27
II-3-3 Comparaison des minuties.....	30
II-4-Conclusion.....	32
Chapitre III : Implémentation d'une solution de reconnaissance d'empreinte digitale	
III-1 Introduction.....	33
III-2 approche de développement.....	34
III-2-1 Approche MatLab Compiler.....	34

III-2-2 Approche Xilinx System Generator.....	34
III-2-3 Approche adaptation de codes.....	35
III-2-3-1 Présentation de FVS.....	35
III-2-3-2 Les organigrammes de FVS (Work Flow)	36
III-2-3-3 Les bibliothèques de FVS.....	39
III-3 Solution SOC.....	39
III-4 L'environnement de développement EDK	41
III-5 Le flux de conception sous EDK	41
III-6 Le processeur soft Microblaze.....	43
III-7 Les entrées /sorties.....	43
III-8 Système de traitement d'images d'empreintes digitales sur SOC.....	44
III-8-1 Création de la plateforme matérielle.....	44
III-8-2 Création de la plateforme logicielle.....	46
III-9 Intégration de FVS.....	46
III-10 Conclusion.....	48
 Conclusion générale et perspectives.....	 49
 Annexe 1.....	 50
Annexe 2.....	60
 Bibliographie.....	 64

Liste des figures

Figure II-1 : forme de l’empreinte.....	13
Figure II-2 : Les points singuliers.....	14
Figure II-3 : Les trois principales classes d’empreinte.....	14
Figure II-4 : Types de minuties (stries en noir).....	15
Figure II-5 : Système de reconnaissance d’empreintes digitales simplifié.....	16
Figure II-6 : Principe de fonctionnement d’un capteur d’empreintes.....	16
Figure II-7 : Le module extraction.....	18
Figure II-8 : Histogramme avant égalisation (a) et après égalisation (b)	19
Figure II-9 : Amélioration par histogramme.....	19
Figure II-10 : Amélioration de l’image.....	21
Figure II-11 : Principe de la méthode auto-adaptative.....	21
Figure II-12 : Image binarisée avec seuillage local.....	22
Figure II-13 : Squelette de l’image binarisée.....	24
Figure II-14 : Les différentes représentations du squelette.....	25
Figure II-15 : Exemples de détermination du type de minutie	26
Figure II-16 : Exemple de détection de fausses minuties.....	27
Figure II-17 : Exemples de minuties détectées.....	28
Figure II-18 : Validation des terminaisons détectées.....	29
Figure II-19 : Définitions associées à une bifurcation.....	29
Figure II-20 : Exemple de borne de limitation.....	31
Figure III-1 : La fenêtre XPS.....	35
Figure III-2 : le flux de conception sous EDK.....	36
Figure III-3 : Architecture du système.....	40
Figure III-4 : Résultats de FVS.....	47

Introduction Générale

Avec le besoin prononcé de techniques de reconnaissance humaines robustes dans des applications critiques (La sécurité, l'économie, la lutte contre la criminalité...), la biométrie s'est placée comme une technologie viable qui peut être intégrée dans des systèmes de gestion d'identité à grande échelle.

Les systèmes de reconnaissance utilisant la biométrie sont essentiellement basés sur le traitement des images. Ils nécessitent des calculs importants et ce pour réduire le temps de traitement qui constitue un facteur déterminant. Aussi, la taille et la consommation d'énergie sont des critères importants. En effet, la reconnaissance par la biométrie est de plus en plus utilisée dans des systèmes mobiles (PC portables, téléphones, PDA's).

Sur le plan matériel, l'apparition des systèmes sur puce (ou SOC) a révolutionné le monde de la microélectronique et en un peu de temps ils se trouvent être une solution adaptée aux contraintes imposés par les systèmes de reconnaissance biométriques.

L'utilisation des circuits programmables est un avantage considérable des SoC's. Ces circuits nous permettent d'ajouter ou d'enlever des périphériques et des ports d'entrées/sorties selon nos besoins sans avoir à modifier ou à ajouter du matériel ce qui peut réduire les coûts. De plus, le principe du co-design qui caractérise les SoC's, nous permet de développer le hardware et le software en parallèle et nous offre une flexibilité dans la conception, ce qui constitue un atout majeur et de grande importance.

L'objectif principal est de concevoir un système de traitement d'image d'empreinte digitale sur une puce et ce, en utilisant l'environnement de développement EDK de Xilinx.

Ce travail est organisé de la manière suivante :

Le premier chapitre est une introduction à la biométrie, il résume les traits biométriques les plus utilisés et justifie le choix de la reconnaissance par empreinte digitale

Le second chapitre présente le processus de traitement des images d'empreinte depuis l'acquisition jusqu'à la partie comparaison en passant par l'amélioration des images. Il présente également les algorithmes choisis et des images pour illustrer les résultats, et cela sous MatLab.

Le troisième chapitre présente l'environnement de développement EDK ainsi que la méthodologie de conception HARD\SOFT de systèmes sur puces. On présente aussi les différentes approches possibles pour réaliser notre système et l'application de la méthode choisie.

Chapitre I

La biométrie

I-1 Introduction :

La biométrie fait référence aux technologies de mesure et d'analyse des caractéristiques physiologiques individuelles comme les empreintes digitales, l'iris, la rétine, la voix, la forme du visage et les dimensions de la main...etc. Une définition conventionnelle de la biométrie est ainsi donnée :

“La biométrie est la science de reconnaissance automatique de l'identité d'une personne en se basant sur des caractéristiques physiologiques ou comportementales mesurables de l'individu.” [1]

Plusieurs remarques importantes peuvent être faites :

- Les termes : *physiologique* et *comportementale* font référence aux propriétés individuelles qui vérifient l'identité d'une personne, en plus d'une information connue par la personne (mot de passe, numéro d'identification personnel...) ou d'un objet détenu par la personne (une clé, une carte...), qui sont deux autres applications de la sécurité.
- Seules les applications, qui associent une *identité* à une caractéristique physiologique ou comportementale, peuvent être considérées comme technologie biométrique. Certaines applications peuvent effectuer une reconnaissance de caractéristiques physiologiques ou comportementales, mais qui ne sont pas directement destinées à la gestion de l'identité (ex : interaction homme | machine). Les définitions et les distinctions entre les termes *vérification*, *authentification*, *reconnaissance* et *identification* vont être discutées plus loin.
- *L'unicité (individualité* de la caractéristique physiologique ou comportementale) est nécessaire pour séparer deux personnes. C'est-à-dire que les caractéristiques associées à deux identités différentes doivent être *suffisamment distinctes* pour pouvoir les différencier. En d'autres termes, la probabilité que deux personnes ou plus, aient suffisamment de similitudes (caractéristiques similaires) pour induire à une fausse correspondance est très faible.

A noter que dans les systèmes réels, bien que les traits physiologiques soient uniques, les caractéristiques mesurées peuvent ne pas l'être.

- Le terme *automatique* indique que la décision de "reconnaître" une personne ne dépend pas de l'homme, mais de la machine.
- Le niveau d'*intrusion* dans la vie privée (Comment la technologie va interférer avec la vie privée de l'utilisateur) va définir le niveau d'acceptance de l'utilisateur vis-à-vis de la technologie biométrique. Le niveau d'acceptance peut dépendre de plusieurs facteurs comme la facilité d'enregistrement, un possible contact entre l'utilisateur et la machine (Contact qui pourrait gêner l'utilisateur) ou encore la révélation d'informations inutiles à l'identification d'une personne (L'analyse ADN peut révéler des détails sur l'état de santé de l'utilisateur).
- La *vulnérabilité* du système est l'ensemble des risques qui compromettent le système biométrique (et les données qui y sont associées) suite à une activité frauduleuse.

I-2 Termes utilisés en biométrie :

Gabarit :

Les systèmes biométriques ne comparent pas directement les images acquises, mais les traitent de façon les rendre plus aptes à la comparaison par ordinateur et à réduire ainsi leur taille de stockage. Ces images traitées sont appelées *Gabarit*. La création du gabarit dépend de l'algorithme qu'utilise le système biométrique. On cite à titre d'exemple le processus de création du gabarit d'une empreinte digitale :

- 1) Capture de l'image de l'empreinte ;
- 2) Traitement de l'image pour faire apparaître les détails (crêtes et sillons) ;
- 3) Etude de la disposition des détails pour en tirer des points caractéristiques (bifurcations, terminaisons, deltas, centre de figures, îlots)
- 4) Mesure de l'emplacement relatif des points caractéristique, "graphique minutia" constituant la base du gabarit de l'empreinte digitale.

Gabarit d'enrôlement :

Il s'agit du gabarit produit lorsqu'un nouvel utilisateur est « enrôlé » (enregistré) dans le système. Ce gabarit est stocké et utilisé comme version authentique à des fins de

comparaison ultérieure. Un gabarit d'enrôlement doit être de haute qualité puisqu'il fait office de norme pour les authentications.

Gabarit d'authentification :

Un gabarit d'authentification est créé chaque fois qu'un utilisateur tente une authentification sur un système biométrique. Contrairement aux gabarits d'enrôlement, les gabarits d'authentification ne sont pas stockés définitivement. Dans le processus d'authentification, on numérise l'utilisateur afin de produire une image, qu'on traite ensuite en vue d'établir un gabarit d'authentification. On compare ensuite le gabarit d'authentification avec le gabarit d'enrôlement afin de déterminer s'ils se ressemblent suffisamment pour qu'il y ait identification ou vérification.

Vérification et identification :

Suivant l'application, un système biométrique pourrait fonctionner en mode vérification ou identification. Dans la vérification, le système valide l'identité d'une personne en comparant sa donnée biométrique acquise, avec celle qui correspond à son identité dans la base de donnée du système. Dans de tels systèmes, un individu qui souhaite être reconnu décline son identité (par un mot de passe, une carte à puce ou un jeton) et le système fait une comparaison « 1 : 1 » pour déterminer si l'identité prétendue est vraie ou fautive. La vérification est généralement utilisée dans la reconnaissance positive, où le but, est de prévenir l'utilisation d'une identité par plusieurs personnes.

Dans l'identification, le système reconnaît un individu en comparant sa donnée biométrique acquise avec toutes les données enregistrées dans la base de donnée du système pour trouver une éventuelle correspondance. Dans ce cas, le système est conduit à faire une comparaison « 1 : N » sans que la personne ait à décliner son identité. L'identification est la composante essentielle des applications de la reconnaissance négative ou le système établit si la personne existe dans la base de données. Le but de la reconnaissance négative est de prévenir l'utilisation de plusieurs identités par la même personne. L'identification pourrait aussi être utilisée dans la reconnaissance positive pour limiter le rôle de l'utilisateur dans la reconnaissance par souci de commodité (le fait de décliner son identité).

Contrairement à la reconnaissance positive, qui peut se faire par mots de passes, clés et cartes d'accès, la reconnaissance négative ne peut se faire que par la biométrie. Le tableau (I-1) [2] résume la différence entre l'identification positive et négative.

Tableau (I-1) : Identification positive et négative.

Positive	Négative
Prouver que je suis une personne connue par le système.	Prouver que je ne suis pas connu par le système.
Prévenir l'utilisation d'une seule identité par plusieurs personnes.	Prévenir l'utilisation de plusieurs identités par une seule personne.
Comparaison un à un.	Comparaison un à "N".
Une fausse correspondance mène à une fausse acceptance.	Une fausse correspondance mène à un faux rejet.
Une fausse non correspondance mène à un faux rejet.	Une fausse non correspondance mène à une fausse acceptance.
Il existe des méthodes alternatives.	Pas de méthodes alternatives.

I-3 Les caractéristiques biométriques :

Plusieurs caractéristiques biométriques sont actuellement utilisées dans plusieurs applications. Chacune d'elles a ses propres avantages et inconvénients, pour cela, le choix d'un trait biométrique pour une application particulière dépend de plusieurs questions relatives à la performance du système qu'on souhaite réaliser. Dans la référence [1], sept facteurs ont été définis pour déterminer s'il est souhaitable d'utiliser un trait biométrique dans une application donnée :

- **L'universalité :** Toutes les personnes qui ont accès à l'application doivent posséder le trait en question.
- **L'unicité :** Le trait donné devrait être suffisamment différent à travers les individus qui composent la population visée.
- **La permanence :** Le trait biométrique d'un individu devrait être suffisamment invariant dans le temps pour respecter l'algorithme de correspondance. Un trait qui change de façon significative dans le temps n'est pas utilisable en biométrie.

- **La mesurabilité** : Il devrait être possible d'acquérir et de numériser le trait biométrique (en utilisant l'outil adéquat pour éviter tout désagrément). En outre, les données extraites devraient être utilisables par le système pour extraire les caractéristiques nécessaires à l'identification.
- **La performance** : L'exactitude de la reconnaissance et les ressources exigées pour réaliser cette exactitude (Algorithmes, matériel...) doivent être imposées par les contraintes de l'application.
- **L'acceptabilité** : Les individus qui vont utiliser le système devraient être d'accord pour présenter leur traits biométriques au système.
- **Mise en échec** : Dépend de la facilité avec laquelle le trait en question pourrait être imité. Aucun trait biométrique ne peut satisfaire efficacement toutes les contraintes (Exactitude, l'aspect pratique, le coût...) imposées par l'application. Autrement dit, aucun trait biométrique n'est idéal mais un certain nombre d'eux sont *admissibles*. Le choix d'un trait biométrique spécifique à une application est établi selon les exigences de l'application et les propriétés du trait biométrique. On donne une brève introduction à certaines des caractéristiques biométriques les plus utilisées ci-dessous :

- Reconnaissance faciale :

Les systèmes biométriques de reconnaissance faciale sont intéressants, car on peut les utiliser à distance, sans la collaboration du sujet. La reconnaissance faciale peut se faire de façon *statique* (ex. une photo) ou bien *dynamique* (ex. camera de surveillance). Elle est basée en générale sur la localisation et la forme des caractéristiques du visage comme les yeux, les sourcils, le nez, les lèvres, et le menton et leurs positions relatives. Les performances des systèmes qui sont sur le marché sont raisonnables, cependant, ils imposent un nombre de restrictions relatives aux conditions d'acquisition de l'image (lumière, angle de vue,...)

L'inconvénient est que si le visage, en théorie, est une base suffisante pour l'identification, dans la pratique, un système de reconnaissance faciale doit, pour

fonctionner correctement, détecter la présence d'un visage dans l'image, le localiser, et reconnaître la personne dans différentes conditions (luminosité, angle d'acquisition). Cela exige l'utilisation d'algorithmes très robustes. Le tableau (I-2) ci-dessous résume les avantages et inconvénients de la reconnaissance faciale :

Tableau (I-2) : Avantages et inconvénients de la reconnaissance faciale

Avantages	Inconvénients
<ul style="list-style-type: none">• Peuvent s'appuyer sur les équipements d'acquisition des images actuels• Peuvent comparer des images statiques, comme des photos de permis de conduire• Seule technique biométrique offerte sur le marché capable de fonctionner sans la collaboration du sujet	<ul style="list-style-type: none">• Les changements dans l'environnement d'acquisition des images (principalement la lumière et l'angle de l'appareil photo) peuvent avoir une incidence sur l'exactitude de la concordance• Les changements physiques peuvent tromper le système• Fortes préoccupations relatives au respect de la vie privée en raison de leurs capacités d'enrôlement et d'identification sans la collaboration des sujets

• La reconnaissance des empreintes digitales :

De part son efficacité, la reconnaissance des empreintes digitales est la technique biométrique la plus utilisée. Une empreinte digitale est la forme des crêtes et sillons sur la surface du bout d'un doigt. Il a été démontré que chaque être humain possède sa propre empreinte et qu'elle est unique, même des jumeaux identiques ont des empreintes différentes. Elle est aussi différente pour des doigts différents de la même personne. De nos jours, les capteurs d'empreintes digitales coûtent moins de 50\$ US et on les rencontre dans un grand nombre d'applications. Cependant, il existe certains inconvénients qui sont résumés dans le tableau (I-3).

Tableau (I-3) : Avantages et inconvénients des systèmes de reconnaissance des empreintes digitales.

Avantages	Inconvénients
<ul style="list-style-type: none">• Technique évoluée et éprouvée• Grande précision• Peuvent être installés dans divers Milieux• Dispositifs ergonomiques et faciles à Utiliser• Permettent d'enrôler plusieurs empreintes (plus grande précision et fiabilité)	<ul style="list-style-type: none">• Enrôlement impossible pour un faible Pourcentage (Facteurs génétiques, environnementaux, professionnels)• Bruit causé par le système d'acquisition• Possible association psychologique à une enquête criminelle• Usure de l'empreinte avec le temps

- Géométrie de la main :

La reconnaissance de la géométrie de la main se base sur un nombre de mesures prises à partir de la main incluant sa forme, la taille de la palme, la longueur et la largeur des doigts. Les systèmes d'authentification qui se base sur la géométrie de la main ont été très utilisés dans le monde (Aéroports de New York, Los Angeles, San Francisco, Toronto, Vancouver...). La technique est très simple, facile à utiliser et pas très chère. Les facteurs environnementaux comme l'humidité de la main et la saleté n'affectent pas la justesse de l'authentification. Cependant, la géométrie de la main n'est pas une caractéristique très distinctive. Pour cela, on ne peut pas l'utiliser dans l'identification d'un individu issu d'une large population. En plus, la géométrie de la main peut varier avec le temps (croissance, port d'une bague). Aussi, la taille des systèmes d'acquisition (relativement grande) exclu leur utilisation dans les systèmes mobiles. Le tableau (I-4) résume les avantages et inconvénients de ces systèmes.

Tableau-4 : Avantages et inconvénients des systèmes de reconnaissance de la main

Avantages	Inconvénients
<ul style="list-style-type: none"> • Technique sûre et bien établie • Généralement perçus comme non Contraignants • Enrôlement et vérification rapides 	<ul style="list-style-type: none"> • Précision restreinte • Le format volumineux de l'appareil peut restreindre ses applications

• Reconnaissance de l'iris :

La reconnaissance de l'iris est considérée comme la technique biométrique la plus exacte. L'iris est la partie colorée de l'œil qui entoure la pupille noire. Son inspection attentive révèle de nombreuses structures détaillées. Ces structures se développent dès le plus jeune âge et sont considérées comme uniques, même chez les vrais jumeaux. Les structures de l'iris resteraient constantes jusqu'à la mort.

La technique de reconnaissance de l'iris n'est pas aussi conviviale que celle de la reconnaissance des empreintes digitales. L'utilisateur doit se prêter à la capture de l'image en plaçant sa tête à un endroit précis et en fixant l'appareil, l'œil grand ouvert, pendant plusieurs secondes. Cela représente une difficulté considérable pour certains utilisateurs, qui ne peuvent pas être enrôlés dans le système en raison de la procédure physique et de l'inconfort qu'elle entraîne. La procédure n'est toutefois pas trop exigeante pour un utilisateur qui jouit d'un état de santé et d'une mobilité raisonnable. Le tableau (I-5) résume les avantages et inconvénients de la reconnaissance de l'iris.

Tableau (I-5) : Avantages et inconvénients de la reconnaissance de l'iris.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Potentiel de très grande précision • Peuvent être utilisés pour l'identification et la vérification • Les structures de l'iris restent stables durant toute la vie. 	<ul style="list-style-type: none"> • L'acquisition des images exige une certaine formation et de la pratique. • L'acquisition des images crée un certain inconfort chez l'utilisateur, ce qui peut empêcher l'enrôlement de certaines personnes.

I-4 Evaluation d'un système biométrique :

Contrairement aux systèmes à mot de passe, où une correspondance exacte entre deux codes alphanumériques est nécessaire pour valider une identité, Un système biométrique rencontre rarement deux échantillons du trait biométrique d'un utilisateur qui aboutissent exactement aux mêmes caractéristiques. Cela est essentiellement dû, aux conditions d'acquisition, au changement des caractéristiques biométriques de l'utilisateur, aux conditions ambiantes et à la variation de l'interaction de l'utilisateur avec le capteur. Ainsi, il est rare que deux caractéristiques provenant du même trait biométrique d'un utilisateur restent exactement les mêmes.

La variation observée dans une caractéristique biométrique d'un individu est dite variation d'intra-classe, alors que la variation entre les caractéristiques biométriques originaires de deux personnes différentes est connue comme variation d'inter-classe. Une caractéristique biométrique utilisable possède une petite variation d'intra-classe et une large variation d'inter-classe.

Le degré de similarité entre deux caractéristiques biométriques est indiqué par un score de similarité. Il est dit score *véritable* ou *authentique* s'il est le résultat de la correspondance de deux traits biométriques d'une même personne. Il est connu comme score *imposteur* s'il est le résultat de la comparaison de deux échantillons issus de personnes différentes. Un score imposteur qui dépasse le seuil η donne une fausse acceptance (ou une fausse correspondance), alors qu'un score authentique qui est en dessous du seuil η est un faux rejet (ou fausse non-correspondance).

Le taux de fausses acceptances (FAR : False Accept Rate) est défini comme la fraction du nombre de scores imposteur dépassant le seuil η . De même, le taux de faux rejets (FRR : False Reject Rate) est défini comme la fraction du nombre de scores authentiques en dessous du seuil η . Le taux de véritables acceptances (GAR : Genuine Accept Rate) est la fraction du nombre de scores véritables dépassant le seuil η . Alors,

$$GAR = 1 - FRR$$

Le fait de modifier la valeur du seuil η change les valeurs du FRR et le FAR, mais il est impossible de diminuer ces deux erreurs simultanément.

A titre d'exemple on peut citer le tableau (I-6) [1] qui compare les taux FRR et FAR de différents systèmes biométriques en se basant respectivement sur, l'empreinte, la face et l'iris et cela dans des conditions bien précises.

Tableau (I-6) : comparaison entre les taux FRR et FAR de différents systèmes biométriques. [1]

Trait biométrique	Condition du teste	FRR	FAR
Empreinte digitale	Rotation, distorsion exagérée de la peau	2%	2%
Visage	Variation de la lumière, intérieur/extérieur, temps.	10%	1%
Iris	Environnement intérieur, plusieurs visites.	0.99%	0.94%

I-5 Conclusion :

Dans ce qui a précédé, on a déterminé que le choix d'un trait biométrique se faisait suivant des critères qu'on a définis. Dans notre application, on se doit d'ajouter le critère de la taille du système car il sera réalisé sur SOC (très petit). Aussi, la disponibilité du capteur à utiliser est un facteur déterminant. Pour ces deux contraintes notre choix se porte sur l'empreinte digitale.

Chapitre II

Traitement de l'empreinte digitale

II-1 Définition :

Une empreinte digitale est la représentation de l'épiderme d'un doigt, elle est constituée d'un ensemble de lignes localement parallèles formant un motif, on distingue les *stries* (ou crêtes, ce sont les lignes en contact avec une surface au toucher) et les *sillons* (ce sont les creux entre deux stries). Les stries contiennent en leur centre un ensemble de *pores* régulièrement espacés.

L'empreinte digitale est unique pour chaque individu et garde *la même forme* tout au long de la vie. Lorsque l'épiderme est altéré, celui-ci se régénère de façon identique. [4]

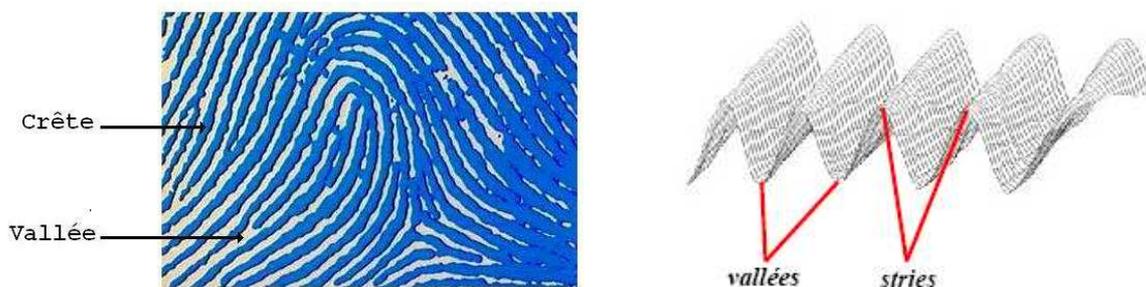


Figure II-1: forme de l'empreinte.

II-2 Caractéristiques des empreintes digitales :

Chaque empreinte possède des points singuliers qui la caractérisent. Il en existe deux types : les points singuliers globaux (les *centres* et les *deltas*) et les points singuliers locaux (les *minuties*).

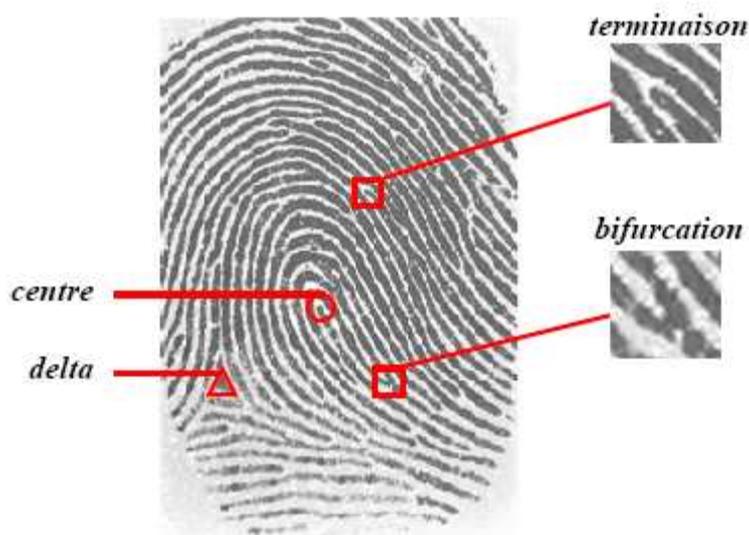


Figure II-2 : Les points singuliers.

II-2-1 Les points singuliers globaux :

On distingue les centres (noyaux) qui correspondent à des lieux de convergences des stries et les deltas qui eux, correspondent à des lieux de divergence. On peut classer les empreintes digitales en catégories selon leur motif général, ce motif est déterminé par la position et le nombre des centres et des deltas. On distingue trois grandes familles d'empreintes digitales, voir (figure III-3) :

- Les boucles (à droite ou à gauche) (*loop*) représentent 65% des empreintes rencontrées.
- Les spires ou tourbillons (*whorl*) représentent 30% des empreintes rencontrées.
- Les arches (*arch*) représentent 5% des empreintes rencontrées.

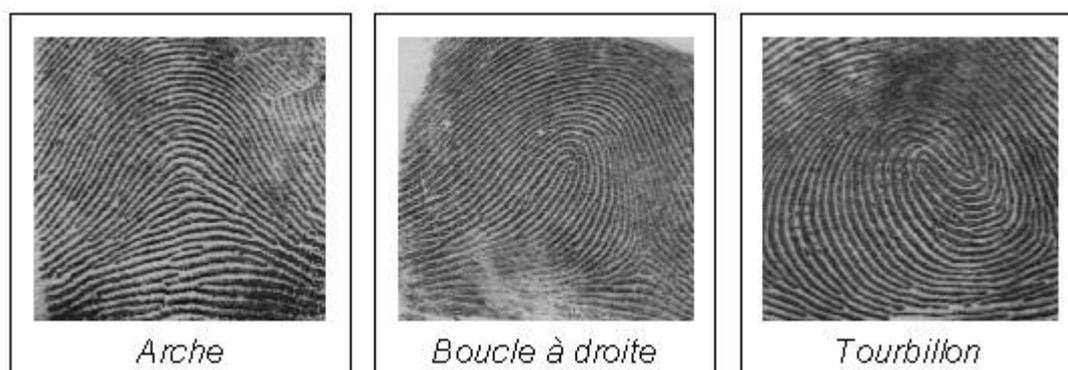


Figure II-3 : Les trois principales classes d'empreinte

II-2-2 les points singuliers locaux :

Ce sont les minuties. Une minutie est un point qui se situe sur le changement de continuité des lignes papillaires. Il en existe plusieurs types, environ treize. Cependant, on ne s'intéresse généralement qu'à deux types à savoir, les terminaisons (fin de strie), et les bifurcations (branches). En effet, on porte un intérêt particulier à ces deux caractéristiques du fait qu'elles sont les plus courantes dans les empreintes et qu'elles permettent de définir d'autres caractéristiques comme le montre la (figure II-4)

- Terminaison à droite ou à gauche (minutie située en fin de strie) (figure II-4 a).
- Bifurcation à droite ou à gauche (intersection de deux stries) (figure II-4 b).
- Lac : assimilée à deux bifurcations (figure II-4 c).
- Île : assimilée à deux terminaisons (figure II-4 d et e). [4]

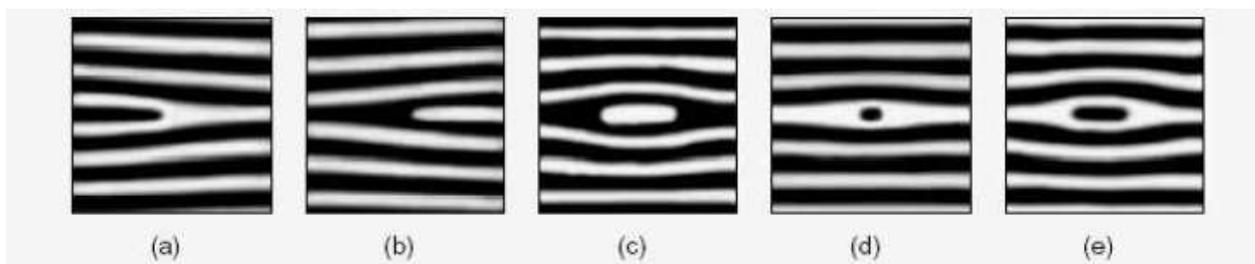


Figure II-4 : Types de minuties (stries en noir).

II-3 Système de reconnaissance d'empreintes :

Un système de reconnaissance d'empreintes digitales est une chaîne de processus qui à partir du doigt d'un utilisateur en entrée renvoie un résultat en sortie. Il est constitué par trois modules : acquisition de l'image, extraction des minuties et leur comparaison.

(Figure II-5). [10]

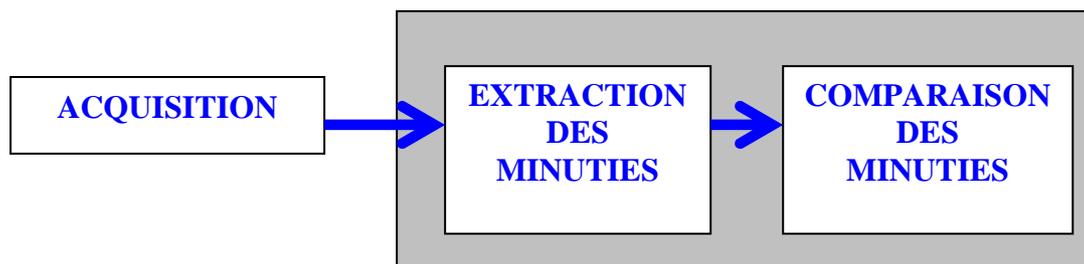


Figure II-5 : Système de reconnaissance d'empreintes digitales simplifié

II-3-1 Acquisition de l'image :

La capture de l'image d'une empreinte digitale consiste à trouver les lignes tracées par les stries.

Le point commun à toutes les technologies utilisées pour la prise d'image d'une empreinte, est que l'image est constituée à partir des points de contact du doigt sur le capteur.

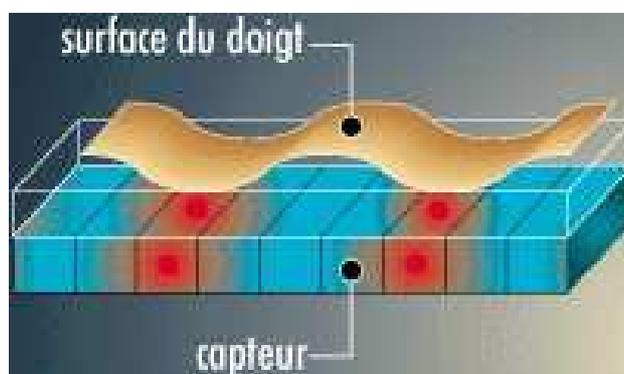


Figure II-6 : Principe de fonctionnement d'un capteur d'empreintes.

Les techniques utilisées pour la mesure sont diverses : capteurs optiques (caméras CCD/CMOS), capteurs ultrasoniques, capteurs de champ électrique, de capacité, de température...

Ces capteurs sont souvent doublés d'une mesure visant à établir la validité de l'échantillon

soumis (autrement dit, qu'il s'agit bien d'un doigt) : mesure de la constante diélectrique relative de l'échantillon, sa conductivité, les battements de coeur, la pression sanguine.

- **Capteur optique** : Il s'assimile à une caméra. Le doigt est apposé sur une platine en plastique dur ou en quartz, qui est en vis-à-vis avec la caméra. Il résiste très bien aux fluctuations de température, mais est gêné par une lumière ambiante trop forte. De plus il est assez volumineux. Son coût est intéressant, et il est intrinsèquement protégé contre les décharges électrostatiques. Il permet d'avoir des images précises et nettes. Ce procédé de capture d'image est le plus ancien après l'encre. Il est fréquemment utilisé particulièrement dans les applications judiciaires pour la qualité de ses images.
- **Capteur en silicium** : Il utilise l'un des quatre effets observables sur les semi-conducteurs : l'effet piézo-électrique, l'effet capacitif, l'effet thermoélectrique et l'effet photoélectrique. Il est en général de très petite taille, d'une durée de vie assez longue, et son coût est très intéressant. Mais, comme tout composant, il est fragile aux décharges électrostatiques et il peut-être détruit si des règles de fabrication et d'installation ne sont pas observées. Ces technologies visent surtout les applications de masse, grâce à une taille réduite et des coûts moins importants que les lecteurs optiques.
- **Capteur à ultra son** : Il utilise une onde ultra sonore qu'il envoie vers le doigt, puis calcule le temps mis par l'onde pour faire un aller-retour. Il est très précis, et hérite des propriétés des ultrasons de traverser certains matériaux. Mais il est volumineux et très coûteux. Il est intéressant pour une population d'utilisateurs très hétérogène. [7]

II-3-2 Module d'extraction des minuties :

La partie extraction se décompose en trois étapes étroitement liées : le prétraitement de l'image, l'extraction et le post-traitement (Figure II-7). Chacune d'elles est très importante car la performance du système va en dépendre. [10]

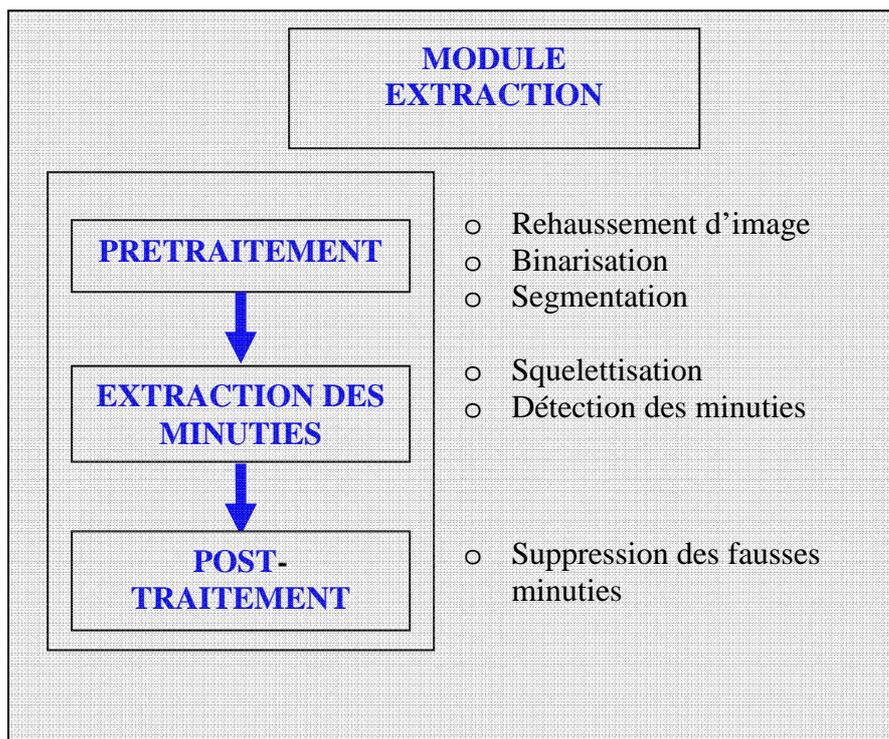


Figure II-7 : Le module extraction

II-3-2-1 Prétraitement de l'empreinte :

II-3-2-1-1 Rehaussement d'image d'empreinte digitale :

Les capteurs n'assurant pas toujours une bonne qualité d'image, Il est nécessaire, pour faciliter les opérations intervenant par la suite, de procéder à des améliorations. Elles consistent à augmenter le contraste de manière à faire apparaître les stries, éliminer les discontinuités et l'effet des pores, qui peuvent engendrer des erreurs de détection.

L'image subit les deux opération suivantes :

- Egalisation d'histogramme;
- En utilisant l'approche fréquentielle auto adaptive [6];

a- Egalisation d'histogramme:

L'égalisation d'histogramme doit augmenter la distribution de valeur de Pixel d'une image (Annexe 1). Après celle-ci, l'histogramme occupe toute la gamme de niveaux de gris, de 0 à 255 et l'effet de visualisation est augmenté voir (Figures II-8 et II-9)

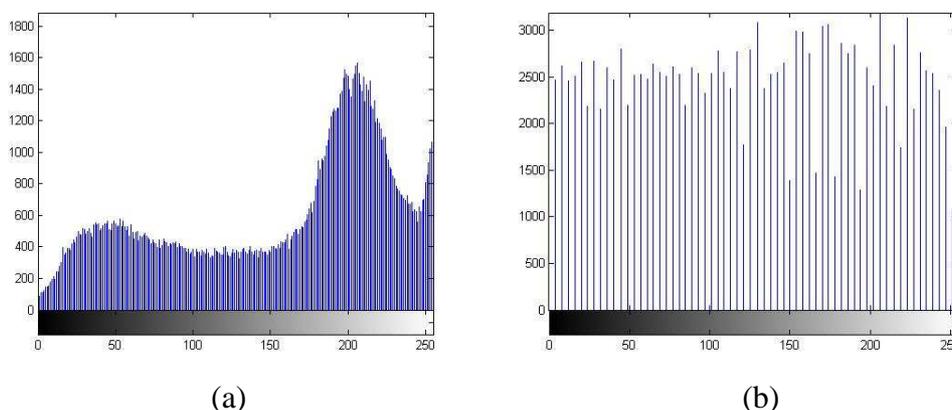


Figure II-8 : Histogramme avant égalisation (a) et après égalisation (b) .



Figure II-9: Amélioration par histogramme (a) Image originale, (b) Image améliorée.

b- Approche fréquentielle auto adaptive :

Principe :

La majorité des algorithmes de traitement de l'image est basée sur l'estimation de l'orientation et la fréquence locales des stries. Ces estimations ont l'obligation d'être très fiables pour assurer un filtrage correct, mais dans le cas d'image très bruitée ces estimations sont plus qu'aléatoire et peuvent provoquer l'effet inverse de celui recherché : à savoir une

dégradation de l'image. Nous utilisons ici la méthode proposée par A.J. Willis et L. Myers ([6]) qui ne nécessite pas ces deux estimations et s'avère plus efficace pour traiter les images de mauvaise qualité.

De manière similaire au calcul d'une carte directionnelle l'image est divisée en blocs de $w \times w$ de telle sorte que les stries du bloc soient localement parallèles, puis la transformée de Fourier Discrète (TFD) est calculée pour chaque bloc (Equ. II-1).

$$f(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \quad (\text{II-1})$$

Pour : $u = 0, 1, 2, \dots, w$ et $v = 0, 1, 2, \dots, w$.

Chaque pixel du bloc fréquentiel obtenu est multiplié par une puissance K de son amplitude (Equ. II-2.). C'est cette simple multiplication qui permet d'obtenir le bloc amélioré.

$$G(u, v) = F(u, v) \times |F(u, v)|^K \quad (\text{II-2})$$

Finalement la partie réelle de la transformée de Fourier discrète inverse (Equ. II-3) est calculée pour chaque nouveau bloc et fournit l'image améliorée.

$$g(x, y) = \Re \left(\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} G(u, v) \cdot e^{j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right)} \right) \quad (\text{II-3})$$

Remarque :

Il est nécessaire de normaliser les éléments de chaque bloc par rapport à l'amplitude maximale, puis de multiplier le résultat par 255 et ce, pour obtenir une image en 255 niveaux de gris

K dans l'équation (II-2) représente une constante déterminée expérimentalement qui est égale à $K=0.45$.

K plus élevé implique une amélioration de l'aspect des stries (remplissage des petits trous qui se trouvent dans les stries). Mais cela provoque un inconvénient, celui de faire apparaître des fausses stries (une terminaison pourrait devenir une bifurcation). (Figure II-10).



Figure II-10 : Amélioration de l'image.

Explication de l'amélioration :

La *FFT* du bloc contient des pics en fréquence suivant les directions dominantes du bloc, dans le cas où le bloc contiendrait des stries parallèles les pics en fréquence vont correspondre à la direction des stries locales au bloc. En multipliant la *FFT* du bloc par son spectre on accentue les stries car le bruit devient négligeable par rapport aux fréquences dominantes comme on peut le voir sur la (Figure II-11). Partant de ce principe, le bloc est amélioré de manière *auto-adaptative* à ses caractéristiques locales sans aucun besoin de calcul de carte directionnelle ou fréquentielle.

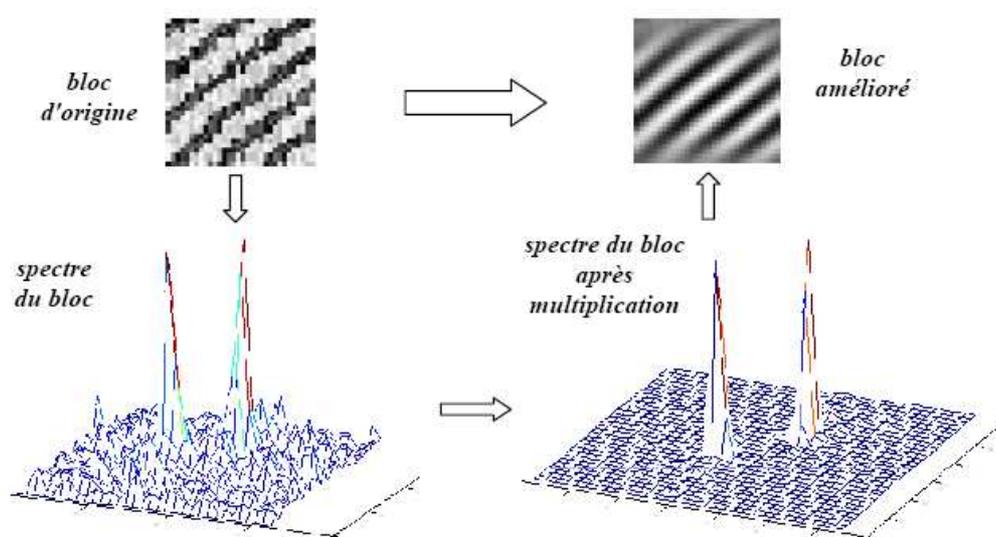


Figure II-11: Principe de la méthode auto-adaptative.

II-3-2-1-2 Binarisation de l'empreinte digitale :

Cette opération consiste à transformer l'image (empreinte digitale) grise de 8 bits à une image à un seul bit (noir (0) et blanc (1)), avec :

Zéro (0) représente des stries et un (1) les vallées.

La méthode qu'on a choisie est la binarisation local, c'est-à-dire que le seuil est calculé localement (seuillage local adaptatif). Dans ce cas l'image est divisée en N blocs et pour chaque bloc la moyenne des pixels du bloc est calculée. Le bloc est ensuite binarisé en fonction de cette moyenne qui va correspondre au seuil. [8]



Figure II-12 : Image binarisée avec seuillage local.

II-3-2-1-3 Segmentation de l'image :

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Il peut s'agir par exemple de séparer les objets du fond. Si l'homme sait naturellement séparer des objets dans une image (compréhension des objets et de la scène), mettre au point des algorithmes de segmentation reste encore un des thèmes de recherche les plus courants en traitement d'images.

La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, parmi lesquelles on peut citer :

- a- Segmentation basée sur les régions.
- b- Segmentation basée sur les contours.

- c- Segmentation basée sur classification ou le seuillage des pixels en fonction de leur intensité.

C'est cette dernière que nous avons retenue pour éliminer les zones qui ne contiennent pas d'information. Il suffit de diviser l'image en blocs de taille N et d'éliminer ceux appartenant au fond de l'image en fonction de leur valeur moyenne. La valeur moyenne locale M_l de chaque bloc est calculée et comparée à un seuil T :

- ◆ Si $M_l < T$, alors on considère que le bloc ne contient aucune information utile et il est éliminé.
- ◆ Si $M_l > T$, alors le bloc contient une partie de l'empreinte, il est conservé.

Le seuil T a été choisi en fonction de la valeur moyenne M_g de l'image globale tel que

$$T = \frac{M_g}{2}$$

Le masque subit ensuite deux traitements:

- ◆ Tout bloc non bruité ayant au moins 3 blocs adjacents définis comme bruités est éliminé (cela correspond à une zone fortement bruitée ou au bord de l'empreinte).
- ◆ L'ensemble des blocs au bord de l'image est supprimé. [13]

II-3-2-2 Extraction des minuties :

II-3-2-2-1 La squelettisation de l'image :

Pour faciliter l'extraction des minuties l'image doit être squelettisée : une suite d'opérations morphologiques d'érosion va réduire l'épaisseur des stries jusqu'à ce que cette dernière soit égale à un pixel tout en conservant la connexité des stries (c'est-à-dire que la continuité des stries doit être respectée, il ne faut pas introduire de trous). [5]

Alors pour chaque pixel P_0 on considère son voisinage immédiat (P_i $i \in [0...8]$) de 8 pixels (Figure IV 5).

Pour l'algorithme de squelettisation on considère les définitions suivantes :

1. P_0 est un point frontière Nord si $P_2 = 0$.
2. P_0 est un point frontière Est si $P_4 = 0$
3. P_0 est un point frontière Sud si $P_6 = 0$
4. P_0 est un point frontière Ouest si $P_8 = 0$.

5. P_0 est un point *8-terminal* si un seul de ses voisins est noir ($\exists! i \in [1..8] P_i = 1$) il s'agit en fait d'une minutie de type *terminaison*.
6. P_0 est un point *8-isolé* si aucun de ses voisins n'est noir ($\forall i \in [1..8], P_i = 0$)
7. P_0 est un point *8-simple* si la *connexité* de ses 8 voisins n'est pas altérée quand on le transforme en pixel blanc.

La squelettisation consiste à répéter les opérations d'érosion suivantes jusqu'à ce que plus aucun pixel ne soit changé:

Etape 1 : tous les pixels noirs vérifiant (1) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Nord).

Etape 2: tous les pixels noirs vérifiant (2) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Est).

Etape 3: tous les pixels noirs vérifiant (3) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Sud).

Etape 4: tous les pixels noirs vérifiant (4) et (7) et ne vérifiant pas (5) et (6) sont changés en pixels blancs (érosion des points frontières Ouest).

La propriété (7) peut être ignorée dans les étapes d'érosion car bien qu'un point unique corresponde à une minutie sa présence à ce stade du traitement est très probablement due à un résidu de bruit, il vaut donc mieux l'effacer. Il est à noter que plus l'épaisseur des stries est importante et plus le processus sera long. La Figure II-13 montre le résultat obtenu à partir d'une image binaire filtrée.

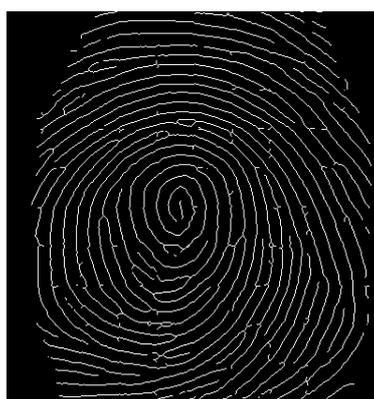


Figure II-13 : Squelette de l'image binarisée.

II-3-2-2-2 La détection des minuties :

Les deux étapes de préparation à l'extraction (binarisation et squelettisation) ont grandement facilité cette phase. En effet nous disposons maintenant d'une image binaire squelettisée: un pixel noir prend la valeur 1, un pixel blanc prend la valeur 0 et la largeur des stries est égale à 1 pixel. Si l'on calcule le nombre de transitions divisé par 2 entre un pixel blanc et un pixel noir pour chaque point du squelette, on obtient le nombre *CN* (*Crossing Number*), partant de ce point, nous pouvons donc déterminer simplement le type d'un pixel voir (Figure II-14).

$$CN(P) = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i-1}|$$

Avec $P_0 = P_8$ et $P_i \in \{0,1\}$

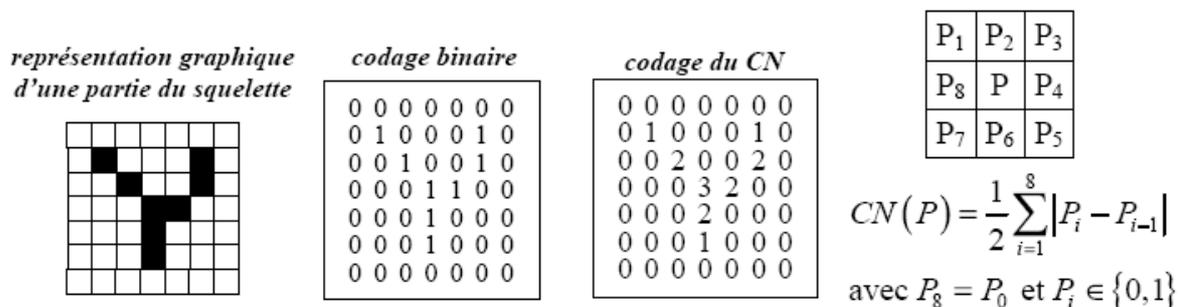
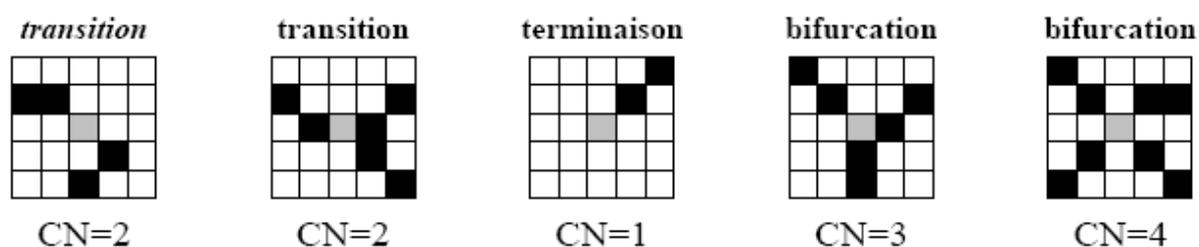


Figure II-14: Les différentes représentations du squelette.

Ainsi pour chaque pixel *P* appartenant à une strie (c'est-à-dire pour chaque pixel ayant une valeur de 1) le calcul de *CN* peut prendre cinq valeurs:

- $CN(P) = 0$: dans ce cas il s'agit d'un pixel isolé et nous n'en tenons pas compte car même si ce type de minutie existe, il est très rare, à ce stade du traitement de l'image. On considère qu'il est probablement dû à un résidu de bruit.
- $CN(P) = 1$: dans ce cas nous avons à faire à une minutie de type *terminaison*.
- $CN(P) = 2$: c'est le cas le plus courant, le pixel se situe sur une strie, il n'y a pas de minutie.
- $CN(P) = 3$: nous sommes en présence d'une *bifurcation* triple.
- $CN(P) = 4$: nous sommes en présence d'une *bifurcation* quadruple. Ce type de minuties étant assez rare il est dû probablement à un bruit dans l'image.



(Dans chaque cas on considère le pixel gris au centre du carré.)

Figure II-15: Exemples de détermination du type de minutie en fonction du calcul de CN .

Bien que l'utilisation du nombre CN facilite grandement la détection, elle provoque aussi la détection d'un nombre très important de minuties (quelques centaines) introduites pour la plupart lors des étapes de binarisation et de squelettisation. De ce fait, on ne peut extraire directement la signature: un traitement supplémentaire est donc nécessaire pour éliminer le plus de fausses minuties possibles.

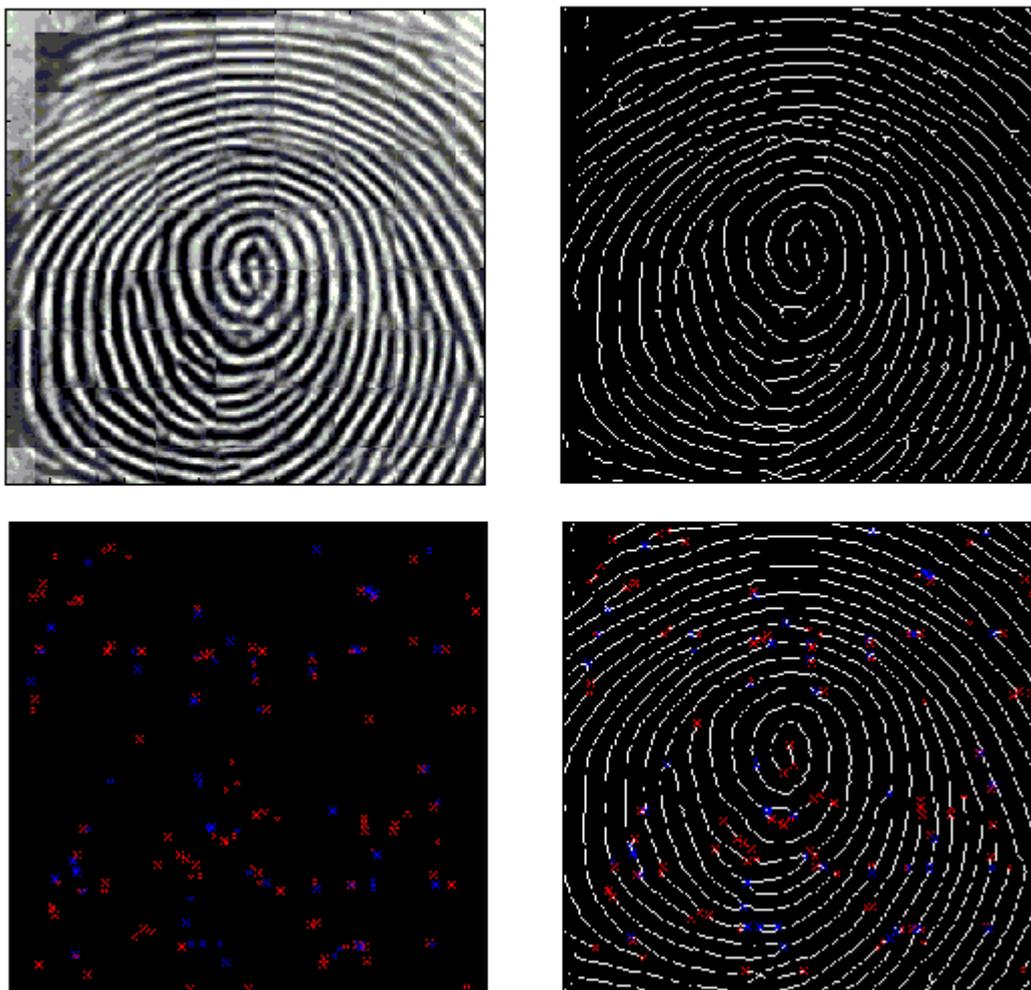


Figure II-16: Exemple de détection de fausses minuties.

II-3-2-2-3 L'élimination des fausses minuties :

Dans la section précédente nous avons vu qu'un traitement supplémentaire est nécessaire pour éliminer la multitude de fausses minuties produites au cours des étapes de binarisation et de squelettisation (Figure II-16). Comme le montre la (Figure II-17) ces fausses minuties sont diverses et variées. L'objectif de ce processus est d'éliminer le maximum tout en conservant les vraies minuties détectées. Pour cela, on utilise des considérations empiriques ([9]-[11]) basées sur le fait que la distance entre deux minuties voisines est toujours supérieure à un certain seuil. En effet, pratiquement il est extrêmement rare de trouver deux vraies minuties très proches, par contre, on a quasiment toujours une concentration locale de plusieurs fausses minuties.

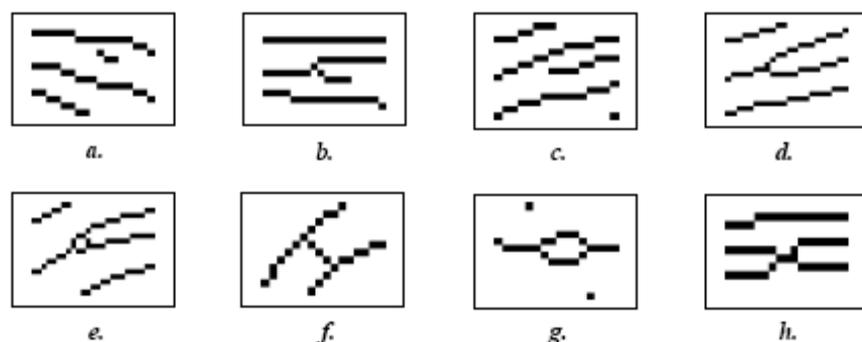


Figure II-17: Exemples de minuties détectées, segment trop court (a), branche parasite (b), vraie terminaison (c), vraie bifurcation (d), triangle (e), pont (f), îlot (g), Segment trop court (h).

Pour faciliter les traitements nous allons travailler non pas sur la représentation binaire du squelette mais sur le codage CN du squelette (Figure II-15, [12]).

- **Le traitement des terminaisons détectées :**

Lorsque l'on détecte un point, $T(x_T, y_T)$ candidat pour le titre de terminaison ($CN(P) = 1$), on vérifie d'abord si celui-ci se situe au bord de l'image car, la très grande majorité des fausses terminaisons sont provoquées par le bord de l'image. Ceci permet d'éliminer de nombreuses fausses terminaisons car les lignes du squelette qui s'arrêtent au bord de l'image provoquent la détection de fausses terminaisons.

S'il existe un bloc adjacent au bloc contenant, $T(x_T, y_T)$ et appartenant à la partie éliminée de l'image (après segmentation), alors on considère que, $T(x_T, y_T)$ est une fausse terminaison et on l'élimine. Cette simple considération permet d'éliminer plus des trois quarts des fausses terminaisons détectées.

Pour les terminaisons restantes T on parcourt la strie qui lui est associée sur une distance maximum K_1 jusqu'à atteindre le point A ($d = TA \leq K_1$, Figure II-18.a). Deux cas de fausses terminaisons sont prises en considération:

- ♦ $d < K_1$ et $CN(A) = 3$: on rencontre une bifurcation avant d'avoir parcouru la distance maximum (Figure II-18.b). C'est le cas d'une branche parasite, le point, $T(x_T, y_T)$ ainsi que la bifurcation A rencontrée sont considérés comme de fausses minuties.

♦ $d < K_1$ et $CN(A) = 1$: on rencontre une terminaison avant d'avoir parcouru la distance maximum (Figure II-18.c). C'est le cas d'un segment trop court, le point, $T(x_T, y_T)$ et la terminaison A rencontrée sont considérés comme de fausses terminaisons.

Pour tous les autres cas, $T(x_T, y_T)$ est validée en tant que *vraie terminaison*.

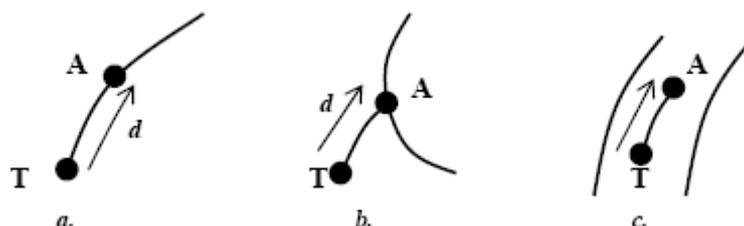


Figure II-18 : Validation des terminaisons détectées, cas d'une vraie terminaison (a), branche parasite (b), segment trop court (c).

Nous pouvons prendre K_1 comme étant la distance moyenne inter stries :

$$K_1 = \bar{d} = \frac{1}{2}(d_{\min} + d_{\max}) = 9 \text{ pixels.}$$

- d_{\min} et d_{\max} sont respectivement les distances inter-stries minimales et maximales.

• **Le traitement des bifurcations détectées :**

Lorsque l'on détecte un point B candidat pour le titre de bifurcation ($CN(A) = 3$), on parcourt les trois stries qui lui sont associées sur une distance maximum de K_1 jusqu'à atteindre trois points A_1 , A_2 et A_3 (Figure II-19).

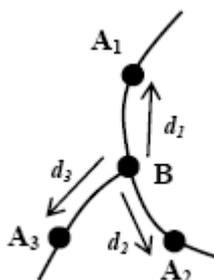


Figure II-19: Définitions associées à une bifurcation lors de la phase de validation.

Plusieurs cas peuvent se produire, ils sont traités dans l'ordre suivant:

◆ $d_1 < K_1$, $d_2 < K_1$ et $d_3 < K_1$: la zone circulaire de centre B et de rayon K_1 contient au moins quatre minuties. On considère alors que nous sommes dans une zone très bruitée (regroupement important) et que B est une fausse bifurcation.

◆ $CN(A_1)=1$ ou $CN(A_2)=1$ ou $CN(A_3)=1$: au moins une des stries mène à une terminaison. On est dans le cas d'une branche parasite, le point B et les terminaisons atteintes ne sont pas validés.

◆ $A_1 = A_2$ ou $A_2 = A_3$ ou $A_3 = A_1$: deux des stries mènent au même point. On est dans le cas d'un îlot (Figure II-17.g). Le point B et la bifurcation atteinte ne sont pas validés. [11][12]

II-3-3 Comparaison des minuties :

Le système de vérification d'identité est basé sur la comparaison de deux ensembles de minuties (fichier "gabarit"), correspondants respectivement à deux doigts à comparer. Pour déterminer si deux ensembles de minuties extraits de deux images correspondent à des empreintes du même doigt, il est nécessaire d'adopter un système de comparaison qui soit insensible à d'éventuelles translations, rotations et déformations affectant systématiquement les empreintes digitales.

A partir de deux ensembles de minuties extraites, le système est capable de donner un indice de similitude ou de correspondance qui vaut :

- 0 % si les empreintes sont totalement différentes.
- 100 % si les empreintes viennent de la même image.

Deux fichiers " *gabarit* " calculées à partir de la même empreinte ne donneront jamais 100 % de ressemblance du fait des différences qui existent lors de l'acquisition de deux images (petites déformations ou déplacements), ils donneront cependant toujours un niveau élevé de similitude. A partir de cet indice de similitude, dire que les deux empreintes sont issues du même doigt devient une question purement statistique. Pour accepter la similitude entre deux " *gabarit* ", il faut arrêter un seuil d'acceptation.

Dans notre travail, on suppose que le capteur est conçu de telle sorte que l'effet des rotations, translations et déformation du doigt soit limité, les différences entre deux fichiers gabarit d'une même image deviennent alors assez minimes.

Après l'opération d'extraction des minuties, un fichier signature est généré. Il est composé de la liste des minuties et est défini de la manière suivante :

- Chaque minutie est définie par son type (*T* ou *B*), ses coordonnées (x,y) et son angle d'arrivée (θ).
- Dans notre démarche on effectue une transformation, c'est-à-dire changer les coordonnées cartésiennes (x,y) par des coordonnées polaires (r, θ) .
- On prend le fichier signature (en coordonnées polaires), et on place des bornes de limitation autour de chaque minutie suivant r et θ ($\pm\Delta r, \pm\Delta\theta$) comme le montre la figure-II-20.

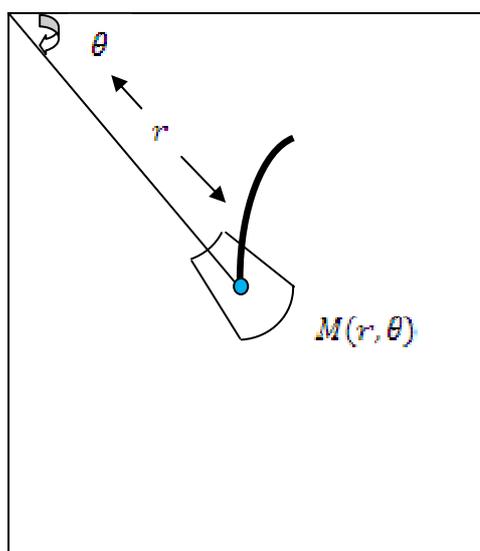


Figure-II-20 : Exemple de borne de limitation.

- Lorsqu'on compare une image à ce fichier signature, on vérifie s'il existe des minuties qui appartiennent à cette borne de limitation, on vérifie aussi son type (terminaison ou bifurcation). Cet algorithme n'est pas très précis car en réalité, il existe toujours des

rotations et des translations. Pour que le niveau de sécurité du système soit élevé, il est nécessaire d'utiliser un algorithme qui prenne en compte les modifications engendrées lors de deux acquisitions différentes.

II-4-Conclusion :

Dans ce chapitre, nous avons vu le processus de traitement d'une image d'empreinte digitale de la première étape qui est l'acquisition jusqu'à la comparaison des images en passant par l'amélioration. Cette dernière est très importante car la précision du système en dépend. Les images et les algorithmes présentés dans ce chapitre ont été validés sous MatLab.

Chapitre III

Implémentation d'une solution de reconnaissance d'empreinte digitale

III-1 Introduction :

De nos jours, nous assistons à l'émergence de nouveaux composants électroniques extrêmement complexes embarquant plusieurs fonctionnalités. De tels composants représentent en réalité des systèmes complets de traitement et/ou transmission de l'information. Ils sont appelés systèmes sur puce ou plus communément SoC (System on Chip). La technologie d'aujourd'hui permet l'intégration de plusieurs processeurs, de plusieurs unités de traitement spécialisées, de la logique programmable jusqu'à 5 millions de portes logiques équivalentes, des bus de communication, des interfaces d'E/S et de la mémoire et tout ceci sur une même puce ne dépassant pas quelques mm² en surface de silicium.

La puissance de calcul offerte par les SoC permet l'exécution en temps réel des algorithmes très complexes à moindre coût, à faible encombrement et à faible consommation. L'intérêt économique suscité par les applications SoC a poussé les créateurs de ces composants à faire une offre très variée répondant à des familles d'applications. Des outils pour la programmation de ces systèmes sur puce sont aussi proposés afin de faciliter leur exploitation.

Dans ce chapitre, on se propose d'étudier ces outils de conception, et de voir quelle serait l'approche la plus adaptée à notre cas. Cependant, toutes ces approches sont basées sur un même principe qui est celui du Co-design, c'est-à-dire développer le matériel en parallèle avec le logiciel.

III-2 approche de développement :

Comme on a vu précédemment, nos programmes ont été écrits et validés sous MatLab, il est à rappeler que le but est d'implémenter une solution de traitement d'empreinte sur SoC. Aussi, trois approches ont été explorées.

- 1- Traduction des codes MatLab en application C en utilisant MatLab Compiler.
- 2- Utilisation Xilinx System Generator.
- 3- Réécrire les algorithmes en C manuellement ou adaptation d'une application C déjà existante.

III-2-1 Approche MatLab Compiler :

A travers cette approche, on souhaite obtenir une application C automatiquement à partir des codes MatLab. Pour cela, on utilise l'outil MatLab Compiler. Il permet de convertir des programmes MatLab en application C ou C++ et qui ne nécessitent pas l'installation de MatLab pour fonctionner. Le compiler prend des applications (M-files, MEX-files ou d'autres codes exécutables MatLab) en entrée et génère des applications autonomes ou des composants software en sortie.

Cette solution paraît idéale pour notre cas. Cependant, afin que l'application générée fonctionne, il est nécessaire de l'utiliser dans le même environnement où elle a été créée. Il aurait été nécessaire d'installer la version Linux de MatLab, générer l'application C, et enfin installer un noyau linux dans le SOC pour pouvoir charger l'application. Cette méthode n'a pas été utilisée car les codes générés ne sont pas optimisés, ils nécessitent beaucoup de temps de calcul.

III-2-2 Approche Xilinx System Generator:

System Generator est un outil MatLab/Simulink qui devient très populaire pour développer des applications de traitement du signal sur FPGA. Constituant une interface entre MatLab/Simulink et ISE de Xilinx, il permet de concevoir et de simuler des systèmes numériques dans un environnement graphique à partir de blocs IP (Intellectual property) et de codes MatLab. Il permet aussi d'importer des projets EDK et de les modifier et/ou de les

améliorer. Il peut être aussi utilisé pour la Co-simulation logicielle (Simulink) – matérielle (FPGA) par communication JTAG.

La solution System Generator paraît aussi être idéale pour notre cas. Cependant, c'est une méthode qui nécessite beaucoup de temps pour maîtriser les outils de conception. Mais cette elle reste intéressante si on veut intégrer des IP cores à notre système.

III-2-3 Approche adaptation de codes:

Vu la contrainte de temps, réécrire les codes MatLab en C est une solution à écarter. Alors, on a opté pour l'approche adaptation de codes. Elle consiste en la recherche de codes C existants de systèmes de traitements d'empreintes digitales, et de les adapter à notre plateforme logicielle. Notre choix s'est porté sur le système FVS (Fingerprint Verification System) de Shivang Patel sur "<http://fvs.sourceforge.net>".

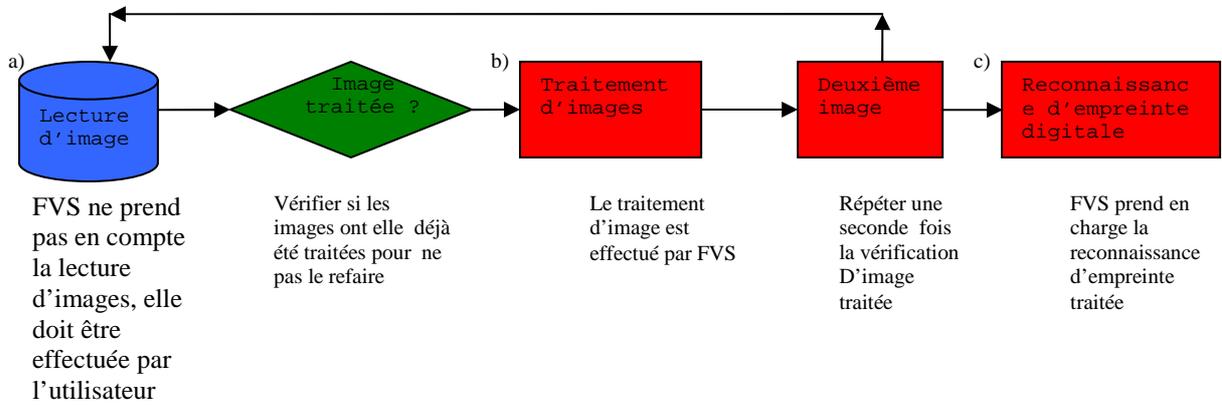
III-2-3-1 Présentation de FVS :

C'est une bibliothèque open source prête à être utilisée, qui permet aux programmeurs d'intégrer la technologie d'empreinte digitale dans leurs applications. FVS est rapide, nécessite peu d'espace mémoire, il est donc idéal pour les applications temps réel. Cependant FVS ne prend pas en compte la lecture d'images, elle doit être effectuée par l'utilisateur. Aussi, certaines modifications doivent être apportées pour l'adapter à l'environnement souhaité. [15]

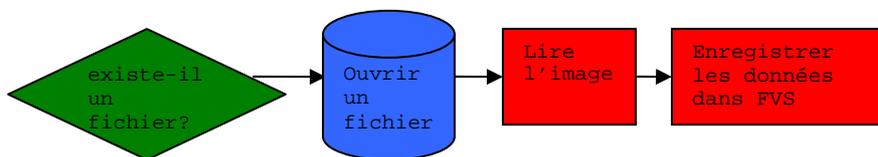
III-2-3-2 Les organigrammes de FVS (Work Flow):

Ces organigrammes ont été conçus par Shivang Patel.

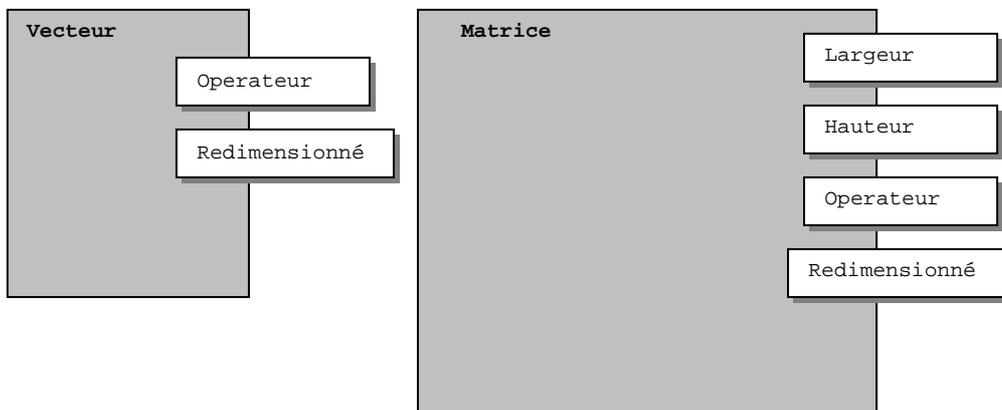
- La librairie de base est disposée sous forme:



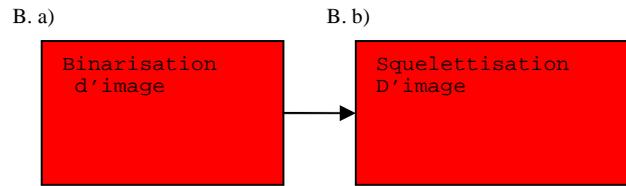
- Organigramme détaillé pour la récupération des images:



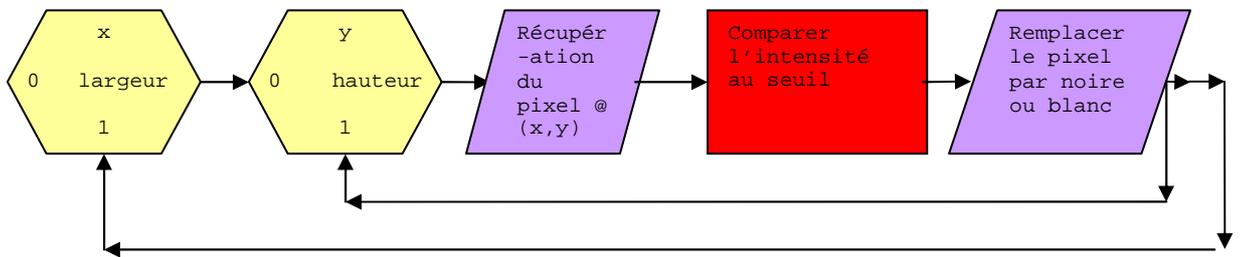
- Le format des images sous FVS:



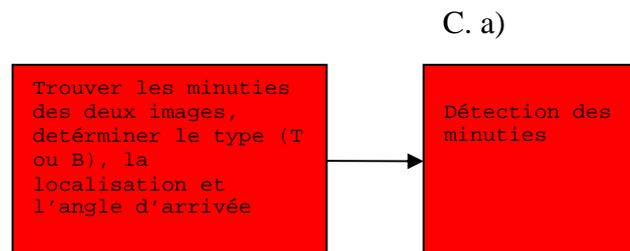
- Organigramme détaillé du traitement d'images :



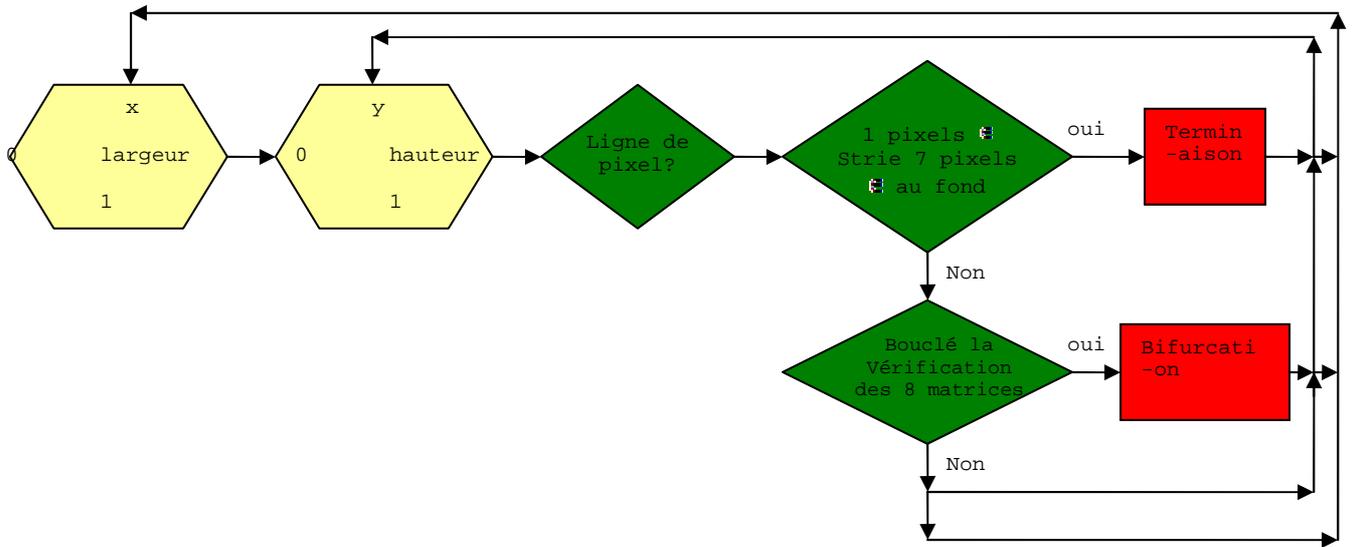
- Organigramme détaillé de la binarisation(B.a) :



- Organigramme détaillé de la reconnaissance d'empreinte



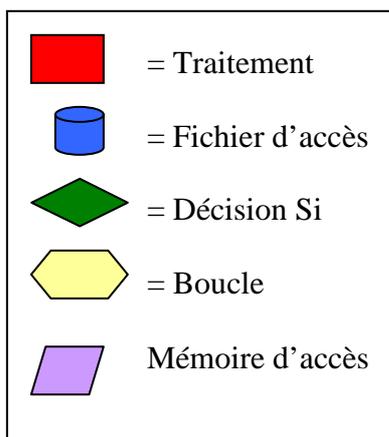
• Organigramme détaillé pour trouver les minuties (partie C.a) :



8 matrices de lignes de bifurcation :

1	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	0	1	1	0	0
0	x	1	1	x	0	0	x	0	0	x	0	1	x	0	0	x	1	1	x	0	0	x	1
1	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1	0	0	0	1	0	0	1	0

1 = strie
0 = fond
x = pixel courant



III-2-3-3 Les bibliothèques de FVS :

Nom du fichier	Type (.c ou .h)	Définition
Config.h	header	Inclut les entêtes de configuration lors de l'étape configuration
Export.h	header	Enregistrement d'une image dans un fichier
Import.h	header	Détection automatique d'un format pour charger une image
fvstypes.h	header	Définit les types utilisés
File.h	header	Manipulation des fichiers. (supprimer, modifier...)
Image.h	header	Information sur l'image et modification (format, taille, couleur...)
minutia.h	header	Objet de détection des minuties
histogram.h	header	Objet histogramme
imagemanip.h	header	Fonctions de traitement d'image
matching.h	header	Algorithmes de comparaison
Img_base.h	header	Manipulation d'images
Fvs.h	header	Inclut tous les header définis dans ce tableau.

III-3 Solution SOC :

Différentes définitions ont été données, dont la plus pertinente est celle qui définit un SOC (System on Chip) comme un système ASIC complexe et indépendant sur une seule puce contenant au moins un processeur. Peu ou pas de périphérique externe viennent compléter le fonctionnement du système.

Un SOC peut être défini comme un IC (Integrated Circuit) complexe qui intègre la majorité des fonctionnalités d'un système sur une seule puce. En général un SOC incorpore au sein de la même puce un ou plusieurs processeurs, de la mémoire, et des accélérateurs

matériels (IP Cores). Il présente des interfaces avec des composants périphériques. La conception SOC combine des composants logiciels **Software** et matériels **Hardware**. Un SOC peut être interfacé avec un environnement externe. Aussi, il peut incorporer des adaptateurs aussi bien analogiques que numériques.

III-4 L'environnement de développement EDK :

L'environnement de développement fourni par Xilinx permettant la configuration des circuits de type FPGA est appelé EDK (Embedded Development Kit). Grâce au principe du Co-Design, cet environnement nous permet d'une part de développer le hardware (configuration du processeur MicroBlaze et de ces périphériques) et d'autre part, de développer le software (les applications) puis de générer un même Bitstream contenant à la fois le matériel et le logiciel qui sera chargé sur le circuit FPGA.

EDK contient un environnement de développement intégré (IDE) appelé Xilinx Platform Studio (XPS), c'est une interface graphique d'EDK qui permet de définir le matériel et le logiciel qui seront implémentés sur le circuit FPGA. EDK contient aussi la plateforme SDK qui peut être utilisée pour développer les applications softwares en langage C ou C++.

- **Xilinx Platform Studio (XPS):**

La fenêtre XPS se divise en plusieurs parties (Figure III-2) :

- 1- une console de transcription qui permet de visualiser les erreurs, les commentaires et les warning présentés lors de la compilation du projet.
- 2- la fenêtre des détails du système (Assembly View) qui montre les différents composants et fichiers du système.
- 3- la fenêtre du diagramme (Block Diagram) qui montre schématiquement les composants et les connections utilisés dans le système.
- 4- la fenêtre d'information (Project Information Area).

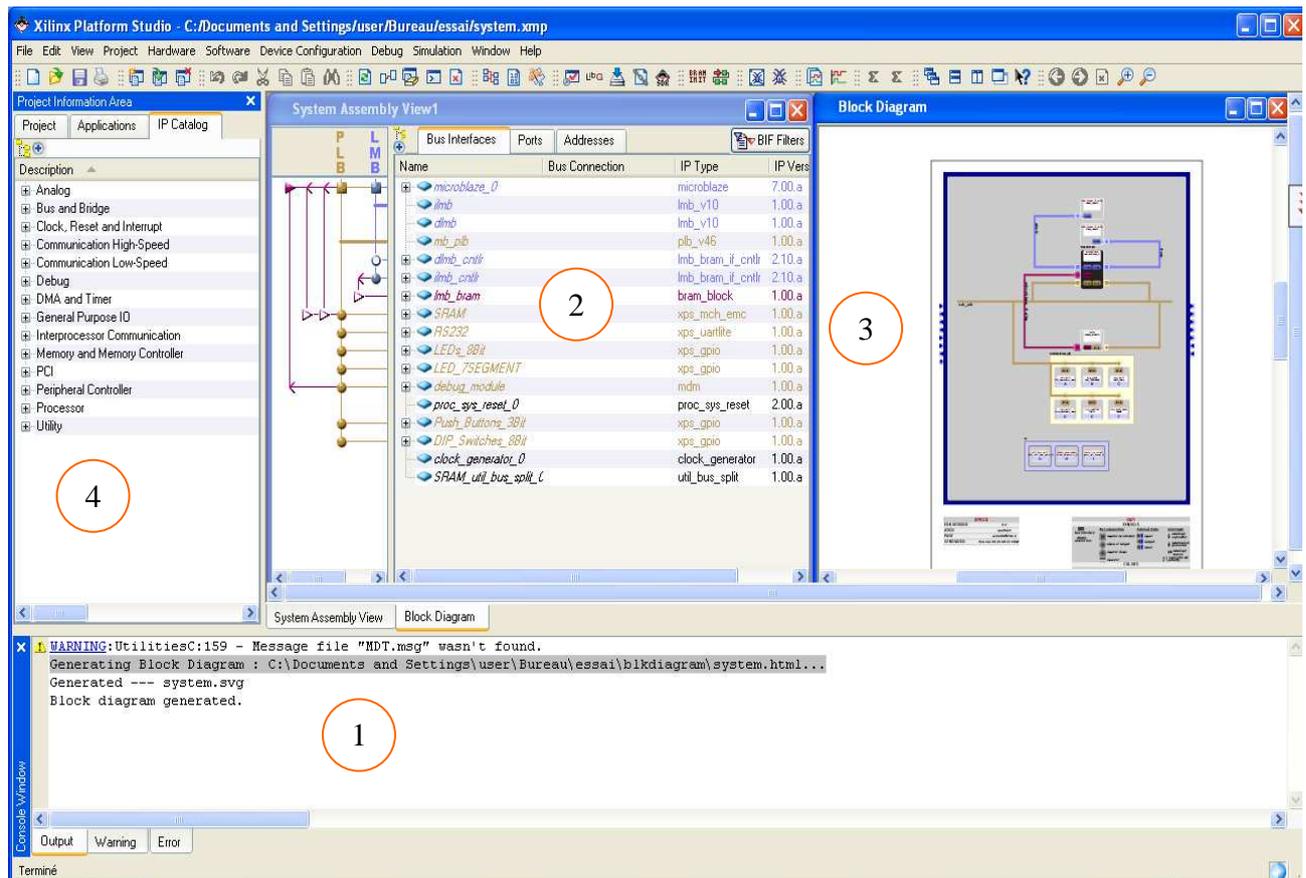


Figure III-1 : La fenêtre XPS.

En plus d'EDK, XPS dépend d'un autre environnement de logiciel intégré (ISE) fourni aussi par Xilinx.

ISE est un produit de Xilinx nécessaire pour implémenter une conception sur le circuit FPGA. L'accès aux différents composants supportés par EDK est impossible sans cet environnement, car de nombreux outils présents dans ISE sont appelés à partir d'EDK pour effectuer diverses tâches.

III-5 Le flux de conception sous EDK :

Le flux de conception d'un projet sous l'environnement EDK est représenté dans la figure suivante :

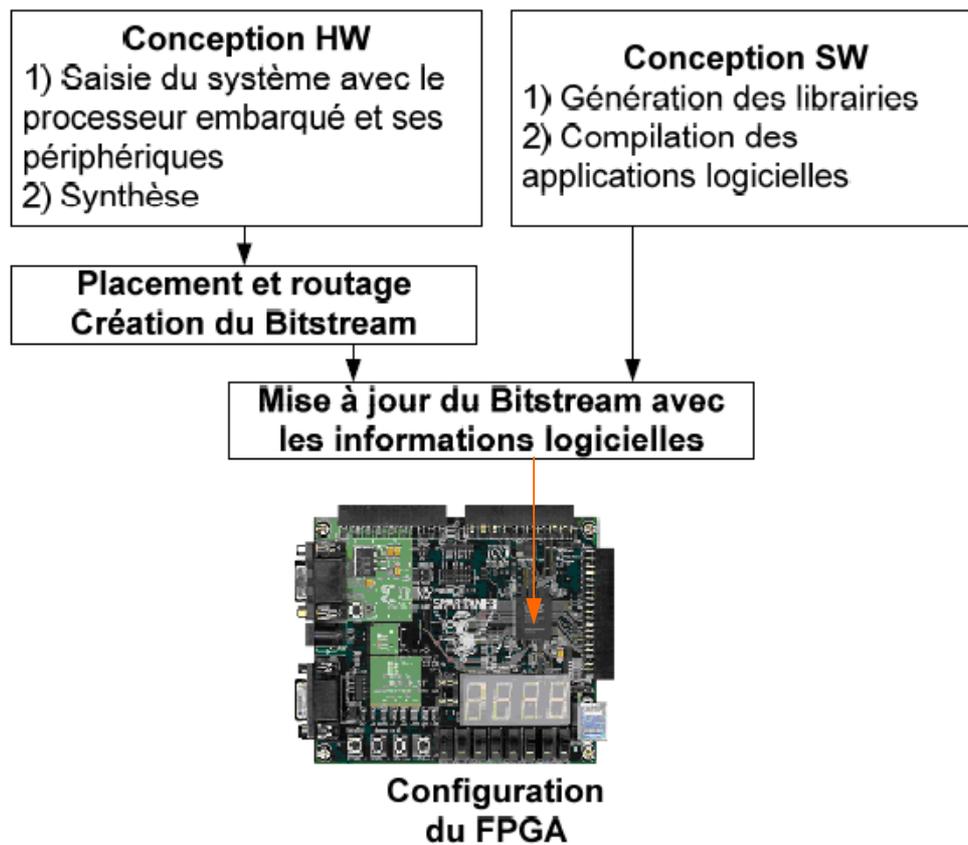


Figure III-2 : le flux de conception sous EDK

L'aspect *co-design* du flux de conception sous EDK nous permet de développer la plateforme matérielle en parallèle avec la plateforme logicielle.

Dans le flux de conception de la partie Hardware, on choisit un processeur IP et les différents périphériques qui lui seront connectés puis on exécute l'outil PlatGen de EDK qui les configure en se basant sur les caractéristiques spécifiées dans le fichier MHS. Une fois la plateforme matérielle spécifiée, une Netlist¹ est synthétisée à partir des codes sources des IP cores propriétaires fournis par l'environnement ISE en langage VHDL. La configuration de la carte et les connexions des différents périphériques avec le circuit FPGA spécifiés dans le fichier de contrainte UCF seront ensuite liés avec la Netlist pour former un fichier *.bit* qui est le fichier de configuration de toute la plateforme matérielle d'un projet sous EDK.

Dans le flux de conception logiciel, le programme de l'application doit être écrit soit en langage assembleur soit en langage C ou C++. Le code source obtenu doit être compilé pour avoir des fichiers en codes objets qui seront liés avec des bibliothèques spécifiques. Ces

¹ Netlist est une forme synthétisée du matériel qui contient les données logiques de la conception et des contraintes

dernières sont générées lors de l'exécution de l'outil PlatGen de EDK pour les spécifications du software qui sont données dans le fichier MSS. A la fin du flux software, on obtient un fichier exécutable.

Le fichier *.bit* obtenu à la fin du flux matériel et le fichier exécutable de l'application obtenu à la fin du flux software sont liés par l'outil DATA2MEM pour former un seul fichier *download.bit* qui configure le système tout entier et qui sera chargé sur le circuit FPGA.

III-6 Le processeur soft MicroBlaze :

Le processeur soft MicroBlaze embarqué est un processeur à jeu d'instruction réduit (RISC²) optimisé pour l'implémentation de Xilinx sur des circuits FPGAs. Développé par Xilinx, le processeur soft MicroBlaze est disponible dans l'environnement EDK de Xilinx sous la forme d'une description faite en HDL (langage de description du matériel : Hardware Description Language). De nombreux périphériques peuvent lui être connectés via le bus OPB tel un contrôleur réseau Ethernet ou un module permettant la gestion du port série.

Conçu selon une architecture dite HARVARD séparant le chemin des instructions de celui des données et permettant un parallélisme dans le transfert des données et des instructions, MicroBlaze réalise l'exécution des instructions en un seul cycle d'horloge grâce à l'utilisation du pipeline.

III-7 Les entrées /sorties :

L'accès à la mémoire peut se faire via trois bus différents :

- **Bus LMB :**

C'est le local Memory Bus, il permet l'accès au Block RAM. Il utilise un nombre minimum de signaux de contrôle et un protocole simple pour assurer que la BRAM est accédée en un cycle d'horloge.

La structure Harvard de MicroBlaze divise le LMB en ILMB pour le transport des instructions, et le DLMB pour les données.

² C'est un jeu d'instruction formé par les instructions les plus simples et les plus fréquemment utilisées. Un processeur RISC se caractérise par : une exécution simple (en un cycle d'horloge) des instructions, une longueur d'instruction uniforme, des modes d'adressage simples, et une couche micro programmée inexistante.

- **Bus OPB :**

C'est le On chip Peripheral Bus, il permet d'accéder aux périphériques on chip. L'OPB est aussi divisé en DOPB (pour les données allant de ou vers les périphériques on chip), et en IOPB (pour les instructions).

- **Le bus FSL :**

FSL est un bus de communication unidirectionnel utilisé pour assurer des communications rapides entre deux éléments (IP-Cores) implémentés sur FPGA. Le MicroBlaze comporte 8 interfaces FSL de type Master/Slave (entrées/sorties). Chaque interface FSL est unidirectionnelle (simplex) et met en œuvre une FIFO (pour stocker les données) et des signaux de contrôle (FULL, EMPTY, WRITE, READ,...). Il met aussi à la disposition du développeur plusieurs fonctions intéressantes dont les plus utilisées sont : "microblaze_bwrite_datafsl" et "microblaze_bread_datafsl". Ces deux fonctions permettent d'échanger des données entre différents MicroBlazes, par exemple, en utilisant la FIFO déjà intégrée dans le bus FSL. Ces deux fonctions sont bloquantes, bwrite se bloque lorsque la FIFO du bus FSL est saturée et bread se bloque lorsque la FIFO est vide. Il doit y avoir un bwrite pour débloquer la lecture. Les communications sur les liens FSL se font très simplement grâce à des instructions prédéfinies. Elles peuvent atteindre les 300 Mo/s à 150 Mhz. [16] [17]

III-8 Système de traitement d'images d'empreintes digitales sur SOC :

La conception de notre système sera faite suivant le principe de co-design comme précédemment expliqué. La partie hard sera conçue indépendamment de la partie soft, puis on intégrera la partie logicielle du système.

III-8-1- Création de la plateforme matérielle :

La plateforme matérielle est créée en utilisant l'outil de génération de plateforme PlatGen. Cet outil a pour entrée le fichier MHS qui définit l'architecture du système, les périphériques, les processeurs intégrés, les connexions du système, la configuration des adresses de chaque périphérique dans le système et les options de configuration de chaque périphérique. PlatGen crée donc une Netlist qui décrit le matériel utilisé en plusieurs formes

(NGC, EDIF). Après l'exécution de cet outil, les outils d'implémentation sur FPGA (ISE) s'exécutent pour compléter l'implémentation du matériel. A la fin du flux ISE, un fichier de configuration (Bitstream) est généré pour configurer le circuit FPGA. Le Bitstream contient l'information pour l'initialisation de la mémoire BRAM sur le circuit FPGA.

Plusieurs périphériques souvent utilisés, sont fournis par Xilinx avec l'outil EDK, cependant on peut définir nos propres périphériques et les inclure dans le fichier MHS.

Il est possible d'éditer son propre fichier MHS, et d'y inclure d'autres périphériques mis à part ceux définis sur EDK, mais cela implique l'écriture (sur ISE) de codes VHDL descriptifs du matériel ajouté.

Dans notre travail, on se propose de réaliser notre système en se basant sur l'architecture général d'un SOC présentée dans la figure III-3 :

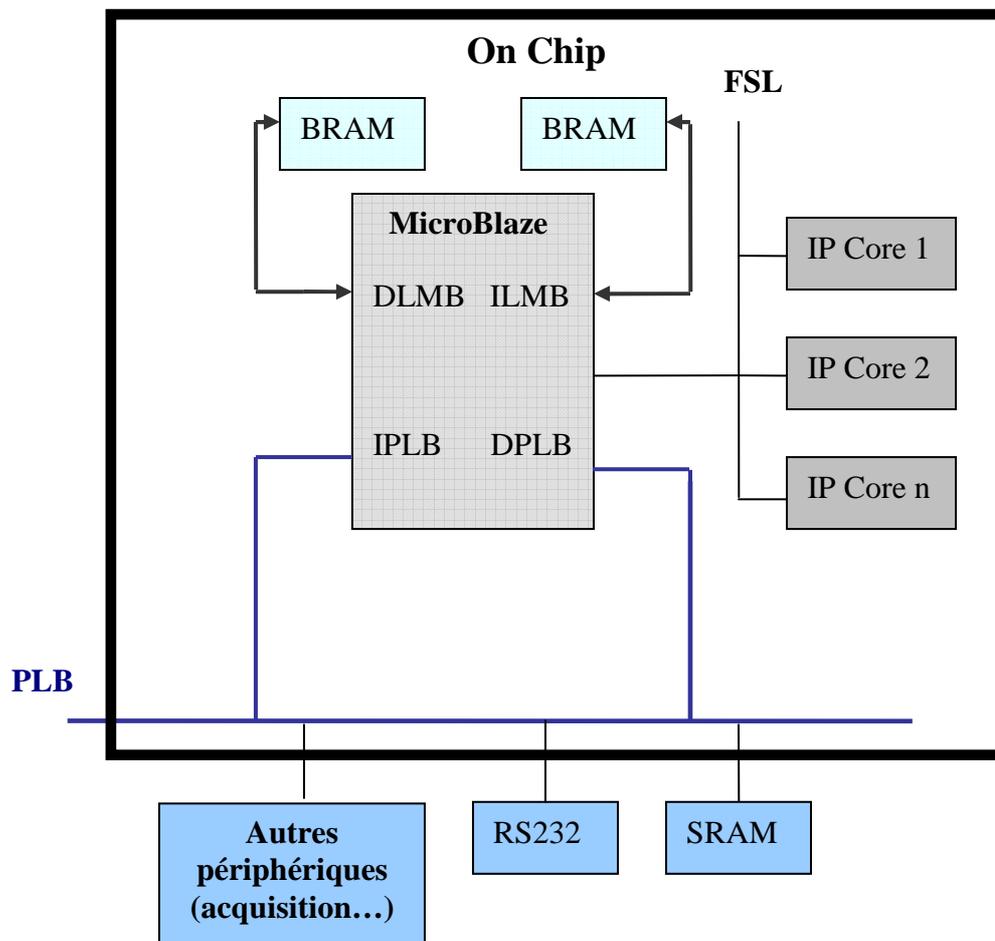


Figure-III-3 : Architecture du système

On utilise Xilinx Platform Studio (XPS) pour réaliser la partie Hard de notre système. L'architecture est essentiellement basée sur un soft processeur MicroBlaze, de la mémoire interne (BRAM) connectée au processeur via le bus LMB, une mémoire externe (SRAM) et un port RS232 connectés au système par le bus PLB. Aucun IP core ne sera intégré à notre système, le FSL ne sera pas utilisé. [16] [17]

La carte que nous possédons est la SPARTAN 3 de xilinx, donc pour des raisons de compatibilité de la carte notre choix s'est porté sur l'utilisation du MicroBlaze et l'environnement EDK de Xilinx. (Incompatibilité de la carte avec le PowerPC). (Annexe 2)

III-8-2 Création de la plateforme logicielle :

La plateforme logicielle est spécifiée par le fichier MSS (Microprocessor Software Specification) qui définit les bibliothèques, les pilotes, les paramètres de personnalisation du processeur, les dispositifs d'entrées/sorties, les routines de traitement d'interruptions et d'autres caractéristiques du software. Le fichier MSS est utilisé comme entrée pour l'outil de génération de bibliothèques (LibGen). Cet outil nous permet de configurer les bibliothèques et les pilotes avec les adresses des périphériques du processeur enfoui.

Les bibliothèques et les pilotes configurés et les programmes sources (d'extension .c ou bien .h) seront compilés grâce à mb-gcc qui est un compilateur GNU adapté à MicroBlaze. Le résultat est un fichier exécutable qui sera chargé dans la BRAM ou la SRAM de MicroBlaze pour être exécuté. [17]

III-9 Intégration de FVS :

Pour intégrer FVS à l'outil de développement d'application soft d'EDK, en l'occurrence SDK (Software development Kit). Il a tout d'abord été nécessaire de charger les bibliothèques ainsi que les codes sources. Il est à noter que FVS utilise la bibliothèque ImageMagick qui permet de lire, de manipuler et d'enregistrer des images.

Après compilation, le système a été bien intégré. Cependant, les fonctions de lecture et d'écriture des images ont été retirées, car on ne peut pas utiliser ImageMagick (Taille très importante, donc elle n'est pas intéressante pour notre système).

. Le fichier « .ELF » qui doit être chargé en mémoire a été créé, mais il a engendré des erreurs. Elles sont dues à la taille du fichier qui ne peut pas être chargé dans les BRAM. Il est nécessaire de configurer notre système de telle sorte qu'il puisse charger le fichier « .ELF » dans la SRAM (Mémoire externe).

Résultat :

Après avoir intégré FVS, nous allons procéder à des tests pour la validation. Comme nous n'avons pas intégré des fonctions de manipulation d'images (ImageMagick), Alors, ces tests seront effectués sur un autre environnement, en l'occurrence Codes Blocks sous Linux. La bibliothèque ImageMagick a été intégrée, les programmes compilés. Nous avons exécuté les fonctions de rehaussement, binarisation, squelettisation et l'extraction des minuties. Nous avons obtenu des images suivantes.



Image rehaussée



image binarisée

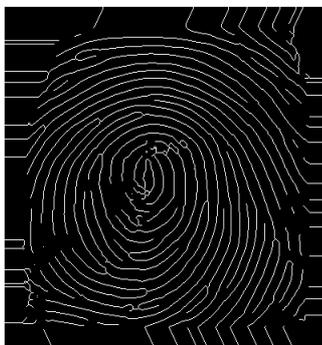
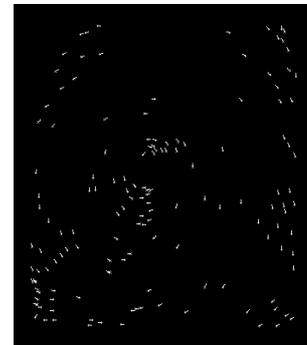


Image squelettisée



détection des minuties

Figure III-4 : Résultats de FVS

Nous observons que le traitement des images a bien été effectué, mais il a engendré certaines distorsions, elles sont dues aux algorithmes utilisés par FVS. Aussi nous avons constaté que l'exécution s'est faite en 10 secondes, ce qui est acceptable.

Des améliorations peuvent être apportées, on peut penser à l'amélioration des algorithmes de traitement, ou encore à l'intégration d'accélérateurs matériels pour réduire le temps d'exécution.

III-10 Conclusion :

Dans ce chapitre divers approches ont été envisagées pour aboutir à notre solution (SoC), mais notre choix s'est porté sur l'approche adaptation de codes (FVS) qui est une approche déjà existante, caractérisée par sa simplicité, sa rapidité et par la taille des codes qui nécessite peu d'espace mémoire.

D'autre part nous avons rencontré quelques difficultés pour utiliser les autres approches, à savoir l'approche MatLab Compiler qui génère des codes qui ne sont pas optimisés, et l'approche Xilinx System Generator qui nécessite beaucoup de temps pour maîtriser ses outils de conception.

Conclusion Générale

Nous avons abordé dans le présent mémoire le développement d'une solution Soc adaptée au traitement de l'empreinte digitale qui reste un cas particulier du thème général de la biométrie.

Pour arriver à notre solution, nous étions emmené à aborder le contexte général, puis le cas particulier de l'empreinte digitale. L'algorithmique a été présentée.

Nous avons fait un choix à priori sachant que notre objectif était la conception d'une solution matérielle.

Pour aboutir à une solution SoC, objectif fondamental de notre projet, diverses approches ont été envisagées. Elles se distinguent soit par leurs niveaux d'abstraction, soit par la taille de la solution obtenue ou par les composants à mettre en œuvre.

La contrainte SoC et la contrainte temps ont orienté notre choix. Les solutions d'abstraction ont été écartées aussi bien pour leur coût que pour leur faible optimisation.

La solution que nous avons proposée est caractérisée par sa portabilité, la faible taille du code ainsi que la faible taille sur circuit sans oublier de mentionner son coût.

Nous pensons que notre solution constitue un bon modèle de solution pour le développement de systèmes biométriques. Elle allie la flexibilité de la solution matérielle et sa facilité d'extension. Les bibliothèques mises en œuvre peuvent être corrigées et enrichies par des solutions exclusivement Soft ou hybride.

Les projections possibles pour son extension sont multiples. Nous citons le développement et la mise en œuvre d'accélérateurs matériels, la personnalisation de la carte support, le développement d'interfaces spécifiques tel que le support réseau. Ce dernier cas peut constituer une excellente solution d'un capteur biométrique réseau.

Annexe 1

I- Notions fondamentales en traitement d'images :

I-1-Image numérique :

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représentée.

La numérisation d'une image est la conversion de celle-ci de son état analogique (distribution continue d'intensités lumineuses dans un plan xOy en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x, y)$ où :

x, y : Coordonnées cartésiennes d'un point de l'image.

$f(x, y)$: Niveau de gris en ce point

Pour des raisons de commodité de représentation pour l'affichage et l'adressage, les données images sont généralement rangées sous formes de tableau I (matrice) de N lignes et P colonnes. Chaque élément $I(x, y)$ représente un pixel de l'image et à sa valeur est associé à un niveau de gris codé sur m bits (2^m niveaux de gris ; $0 = \text{noir}$; $2^m - 1 = \text{blanc}$). La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur.

$$I = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0, P-1) \\ f(0,0) & f(0,0) & \dots & f(0,0) \\ \dots & \dots & \dots & \dots \\ f(0,0) & \dots & \dots & f(0,0) \end{pmatrix}$$

I-2-Pixel :

Abréviation de « Picture Element », mot anglo-américain de **pix** pour pics, abréviation de « picture » (image) et de **el** pour « element ». Élément d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une

quantification. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image.

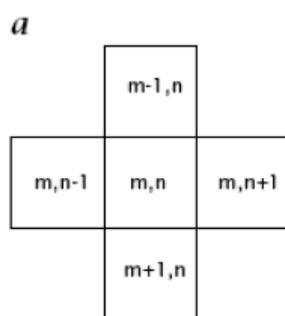
Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : rouge (R), vert (V) et bleu (B).

I-3-Voisinage d'un pixel :

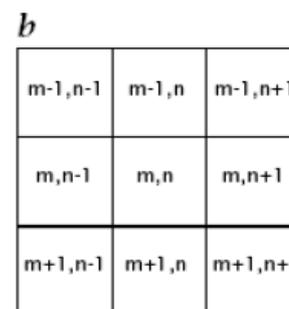
Le voisinage d'un pixel est composé de tous les pixels qui l'entourent immédiatement. Si p est un pixel d'une image D , alors le voisinage de p est le plus petit sous-ensemble de D qui contient p .

Dans une image numérique, on distingue deux types de connexités relatives au voisinage utilisé : la 4-connexité (Figure I.1.a) et la 8-connexité (Figure I.1.b).

Le voisinage d'un pixel (i, j) est dit 4-connexe s'il est formé des quatre pixels de coordonnées spatiales $(i+1, j)$ $(i, j+1)$ $(i-1, j)$ $(i, j-1)$ et il est dit 8-connexe s'il est formé des pixels de coordonnées spatiales $(i+1, j)$ $(i, j+1)$ $(i-1, j)$ $(i, j-1)$ $(i+1, j+1)$ $(i+1, j-1)$ $(i-1, j+1)$ $(i-1, j-1)$.



a) Voisinage 4-connexes



b) Voisinage 8-connexes

Figure I.1 : Voisinage d'un pixel.

I-4-Image en niveaux de gris :

Dans une image en niveaux de gris, la couleur d'un pixel peut prendre des valeurs allant du noir au blanc, en passant par un nombre fini de niveaux intermédiaires. Dans une telle image les intensités du rouge, du vert, et du bleu de chaque pixel sont égales.

En général, les images en niveaux de gris renferment 256 teintes de gris (image à 256 couleurs), simplement chacune de ces 256 couleurs est définie dans la gamme des gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.

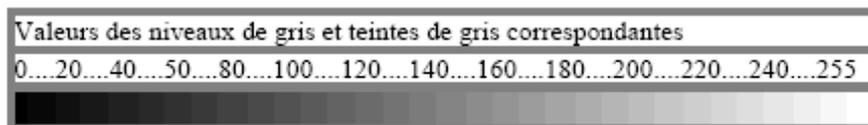


Figure I.2 : Valeurs des niveaux de gris et teintes de gris correspondantes.

I-5-Image binaire :

Ces images sont constituées de points (pixels) qui ne peuvent avoir que deux états : noir ou blanc. On les appelle des images « au trait ». On trouve parmi elles des pages de texte, des signatures, des plans, des dessins, etc.

I-6-Histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. (Figure I.6)

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

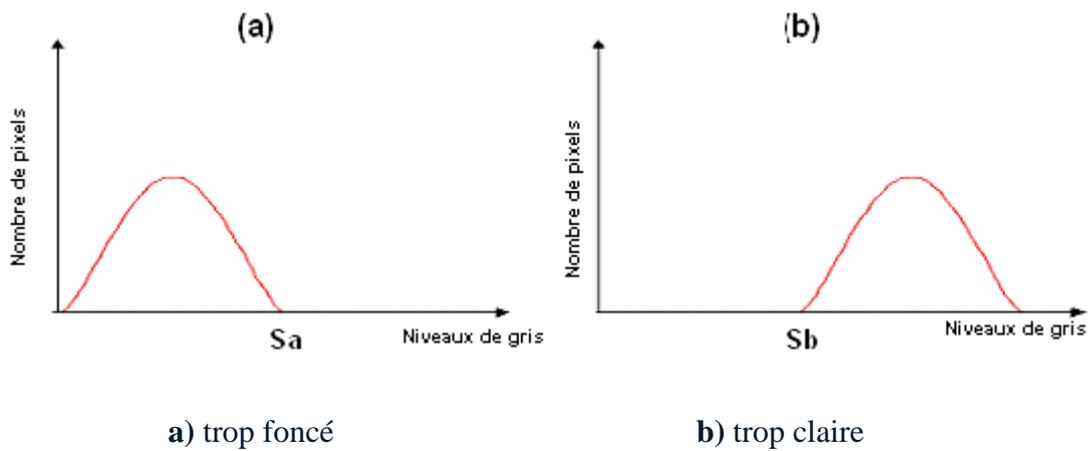


Figure I.3 : Histogramme d'image.

I-7-Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L_1 - L_2}{L_1 + L_2}$$

I-8-Bruit dans l'image :

Le bruit dans une image est défini comme étant un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins, il affecte la qualité de l'image et il provient soit du dispositif d'acquisition (scanner, caméra, amplification, quantification...) soit de la scène elle-même (poussières, rayures...).

II- Traitement des images :

II-1 Binarisation et seuillage :

Les méthodes de binarisation consistent à transformer une image de niveau de gris en une image noire et blanche.

Soit f la fonction de binarisation et $i(x, y)$ un pixel de l'image, on a : $f(i(x, y)) \in \{0,1\}$.

Les méthodes de binarisation se divisent en deux classes : globales et locales.

-*Les globales* : calculent un seul seuil pour toute l'image. Les pixels ayant un niveau de gris plus foncé que le seuil sont mis à noir et les autres à blanc.

-*Les locales* : calculent un seuil pour chaque pixel en fonction de l'information contenue dans son voisinage.

II-2 Augmentation de contraste :

Une famille de traitements a pour objectif de donner à l'image un plus grand contraste. Ces méthodes procèdent de trois façons différentes :

1. Par étirement d'histogramme : on cherche à exploiter au mieux la dynamique de l'histogramme, et cela en utilisant toute l'échelle de gris disponible.
2. Par filtrage passe-haut de l'image : l'idée de base est ici de réduire l'importance du terme continu et des basses fréquences.
3. Par des méthodes locales : ces méthodes modifient localement l'histogramme pour conserver toujours une bonne dynamique, même dans des zones de fort contraste.

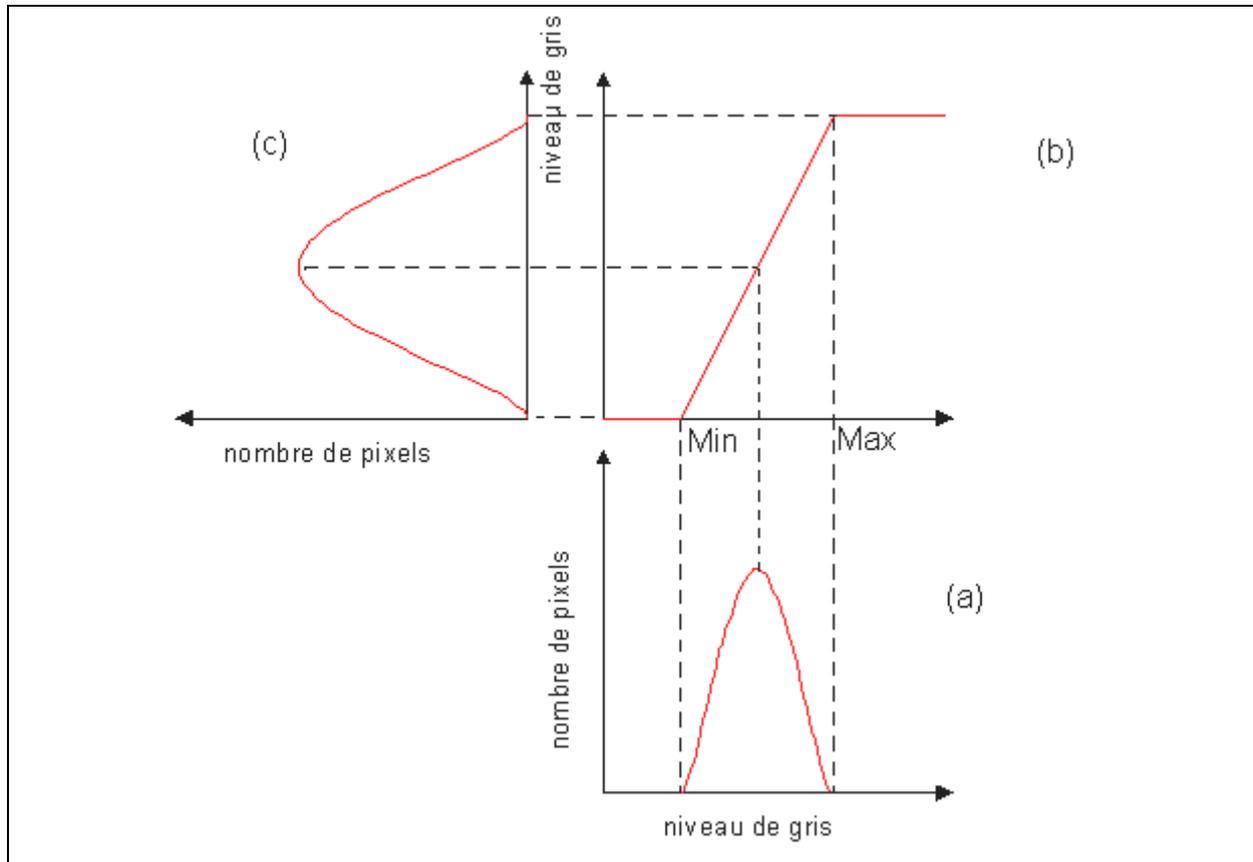
II-2-1 Etirement d'histogramme :

L'histogramme d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (ou couleur) dans l'image. Or parfois certaines images apparaissent trop foncées (les niveaux de gris de l'image sont tassés vers le haut de l'échelle), ou trop claires (les niveaux de gris de l'image sont tassés vers le bas de l'échelle), ou bien peu contrastées (les niveaux de gris de l'image sont regroupés dans un intervalle étroit).

De ce fait, afin d'améliorer l'image on est ramené à modifier son histogramme, qui a pour but de redistribuer les niveaux de gris de l'image afin de leur faire occuper toute la bande de nuance possible. (Figure II-1)

Cette méthode est basée sur les *transformations ponctuelles* d'intensité. C'est-à-dire, qu'à tout pixel d'intensité a_s on associe une intensité $a'_s = T(a_s)$

Du fait de leur caractère ponctuel, les méthodes de transformation d'histogramme n'affectent pas la forme des régions. Elles en modifient uniquement l'apparence visuelle.



a)- Histogramme de l'image originale.

b)- Fonction de transformation.

c)- Histogramme étiré.

Figure II-1 : Principe de la modification d'histogramme.

II-2-2 Filtrage passe haut :

Un filtre « passe haut » favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste. Un filtre « passe haut » est caractérisé par un noyau

comportant des valeurs négatives autour du pixel central. Soit l'image $I(x, y)$ et le masque

$$h = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \text{ l'image filtrée résultante est } I'(x, y) = I(x, y) * h$$

Ce pendant, des effets secondaires au traitement peuvent apparaître comme l'augmentation du bruit granuleux.

II-2-3 Suppression des bruits :

C'est l'un des sujets les plus délicats du traitement des images. De nombreuses méthodes lui ont été consacrées, tout d'abord très intuitives, mais progressivement de plus en plus complexes.

Nous verrons tout d'abord les approches linéaires, non linéaires puis les méthodes morphologiques.

II-2-3-1 Filtres linéaires :

La transformation de chaque pixel est le fruit d'une combinaison linéaire des pixels voisins.

Ces opérateurs sont caractérisés par leur réponse impulsionnelle $h(x, y)$.

La relation entrée-sortie est décrite par l'équation de convolution :

$$C(x, y) = A * h(x, y) = \iint A(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \quad (\text{II-1})$$

Dans le cas d'une image numérique, l'équation (1) sera remplacée par relation de convolution discrète :

$$D[i, j] = \sum_m \sum_n h[m, n] A[i - m, j - n] = \sum_m \sum_n A[m, n] h[i - m, j - n] \quad (\text{II-2})$$

Les filtres linéaires les plus utilisés :
 -Le filtre moyen.
 -Le filtre gaussien.

- **Le filtre moyen :**

L'idée est de réaliser une moyenne des niveaux de gris autour du pixel central. Pour cela, on peut utiliser un masque du type :

$$\frac{1}{9} * \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

La taille du masque est un paramètre variable. Plus le masque sera de grande dimension, plus l'effet du filtrage sera fort. De manière générale, on essayera de réaliser un filtrage isotropique (même effet dans toutes les directions). Pour cela, le voisinage considéré devra avoir une forme circulaire :

$$\frac{1}{29} * \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Bien que cette méthode soit très simple à mettre en oeuvre, elle possède un inconvénient majeur : le filtrage introduit un effet de flou, les contours sont dégradés.

• **Filtre gaussien :**

Il représente un filtre linéaire parmi les plus courants, tant par sa facilité de mise en oeuvre que par la bonne qualité de ses résultats. Ce filtre tire son nom de la valeur de ses coefficients qui sont ceux d'une courbe de Gauss à deux dimensions.

Un filtre gaussien consiste en la convolution d'une image I_B avec une gaussienne $G(x, y, \sigma)$:

$$I_F = G \otimes I_B \tag{II-3}$$

$$\text{Avec : } G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{II-4}$$

Où G est un masque carré dont les coefficients sont les éléments discrétisés de la gaussienne. Le principal avantage du filtre gaussien est qu'il présente le meilleur compromis entre les localisations spatiales et fréquentielles.

II-2-3-2 Filtres non linéaires :

Ce type de filtre s'oppose au précédent dans sa dénomination car il n'est pas le résultat d'une combinaison linéaire de pixels, les pixels voisins interviennent suivant une loi non linéaire.

Le domaine du filtrage non linéaire est extrêmement vaste, l'exemple le plus classique de filtre non linéaire est le filtre médian de Tuckey.

• Filtre médian

Son principe est celui de la (figure II-5) et il peut être décomposé selon les étapes suivantes :

Pour chaque pixel de l'image

- On classe les pixels voisins du pixel courant (compris dans la fenêtre) par valeurs croissantes.
- On prend la valeur médiane des pixels classés et on l'affecte au pixel courant.

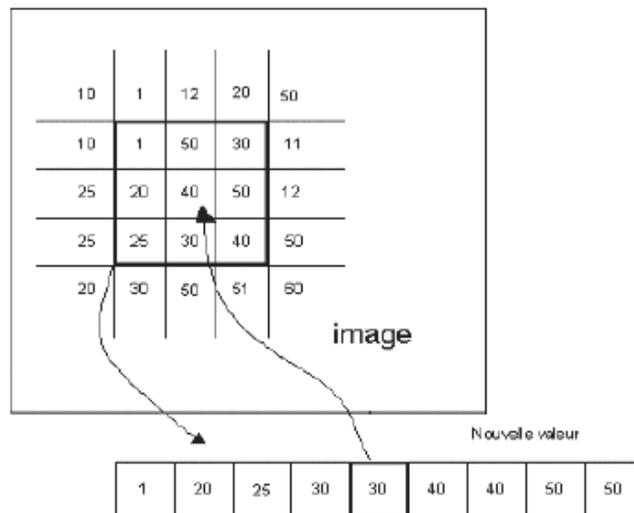


Figure II-5 : Principe de fonctionnement du filtre médian.

II-2-3-3 Filtres morphologiques :

Ce type de filtrage est réalisé par une combinaison de traitement linéaire et non linéaire et il est appliqué à des images binaires, chaque pixel de l'image peut être codé comme un élément logique donc à valeur *vraie* ou *fausse*, ainsi il est l'objet d'une transformation par une fonction logique plus ou moins complexe.

Parmi les opérateurs morphologiques (méthodes visant à améliorer les images binaires) nous citons : la dilatation, l'érosion, l'ouverture et la fermeture.

- **Dilatation**

Elle permet d'éliminer les pixels blancs isolés. On effectue le **ET** logique des huit voisins du pixel considéré.

La dilatation élimine les tâches (les trous) blanches dans les zones noires mais ajoute des pixels noirs au contour des objets présents dans l'image. Donc la taille des objets dans l'image augmente.

- **Erosion**

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image, on effectue le **OU** logique des huit voisins du pixel considéré.

En appliquant une érosion, ces tâches noires peuvent être éliminées mais la taille des objets présents dans l'image diminue car l'érosion enlève des pixels du contour, entraînant une déformation de certains objets.

- **Ouverture**

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet de retrouver la taille normale des objets de l'image.

- **Fermeture**

La fermeture est l'opération inverse de l'ouverture, qui consiste à faire subir à l'image une dilatation suivie d'une érosion. Elle permet aussi de retrouver la taille normale des objets de l'image.

Annexe 2

Présentation générale de la SPARTAN 3 :

La carte de control est fournit selon un kit comprenant la carte les câbles de programmations ainsi qu'un module d'alimentation. Ce kit fait office de plateforme de développement "tout en un" universelle spécialement conçue pour l'apprentissage rapide des techniques de conception numérique.

De part la présence de son FPGA très largement dimensionné (près de 200 K portes) et de ses dispositifs de commandes et de visualisation divers (boutons-poussoirs, afficheurs, Leds, port PS2, Port VGA...), cette platine convient tout aussi bien pour la réalisation d'applications de décodage logique très simple comme pour la mise au point de réalisations extrêmement complexes et puissantes. L'ensemble est livré avec un bloc d'alimentation, un câble de programmation "JTAG" permettant de programmer votre application en mémoire Flash non volatile.

Caractéristiques principales du kit :

- Base de développement FPGA complète avec câble de programmation "JTAG" livré
- Equipé d'un FPGA Spartan-3 avec 216 Kbits de bloc RAM et horloge interne jusqu'à 500 MHz
- Oscillateur 50 MHz inclus - support pour second oscillateur
- Plate-forme flash 2 Mbits (XCF02S) intégrée à la carte
- Mémoire SRAM (256 Kb x 32) intégrée à la carte
- 3 connecteurs d'extensions inclus sur la carte
- 4 afficheurs 7 segments à Leds
- 9 Leds
- 8 interrupteurs et 4 boutons-poussoirs
- Port série, VGA et PS2
- 3 régulateurs de tension (3,3 V / 2,5 V et 1,2 V)
- Livré avec câble de programmation (prévoir une alimentation de : +5 V)

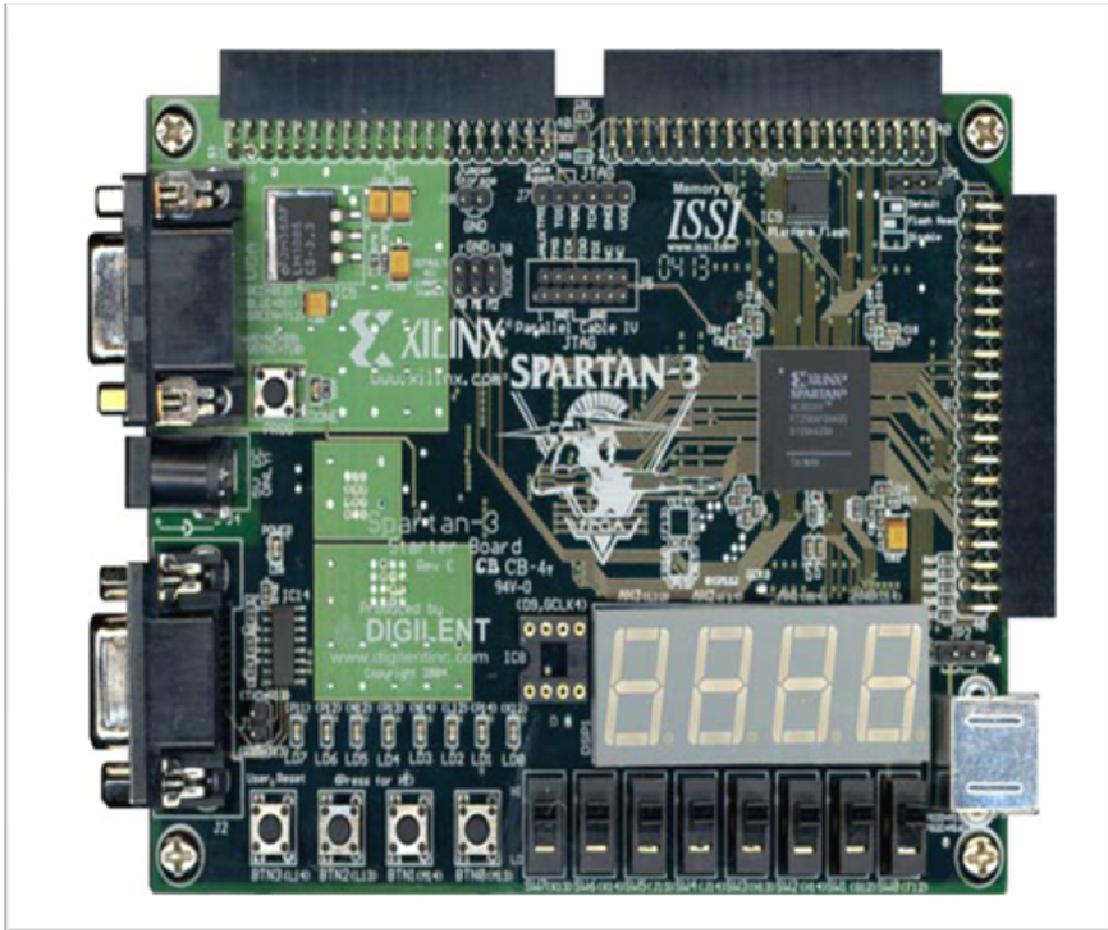


Figure: la Spartan 3 starter board

Composants essentiels de la Spartan 3 Starter board :

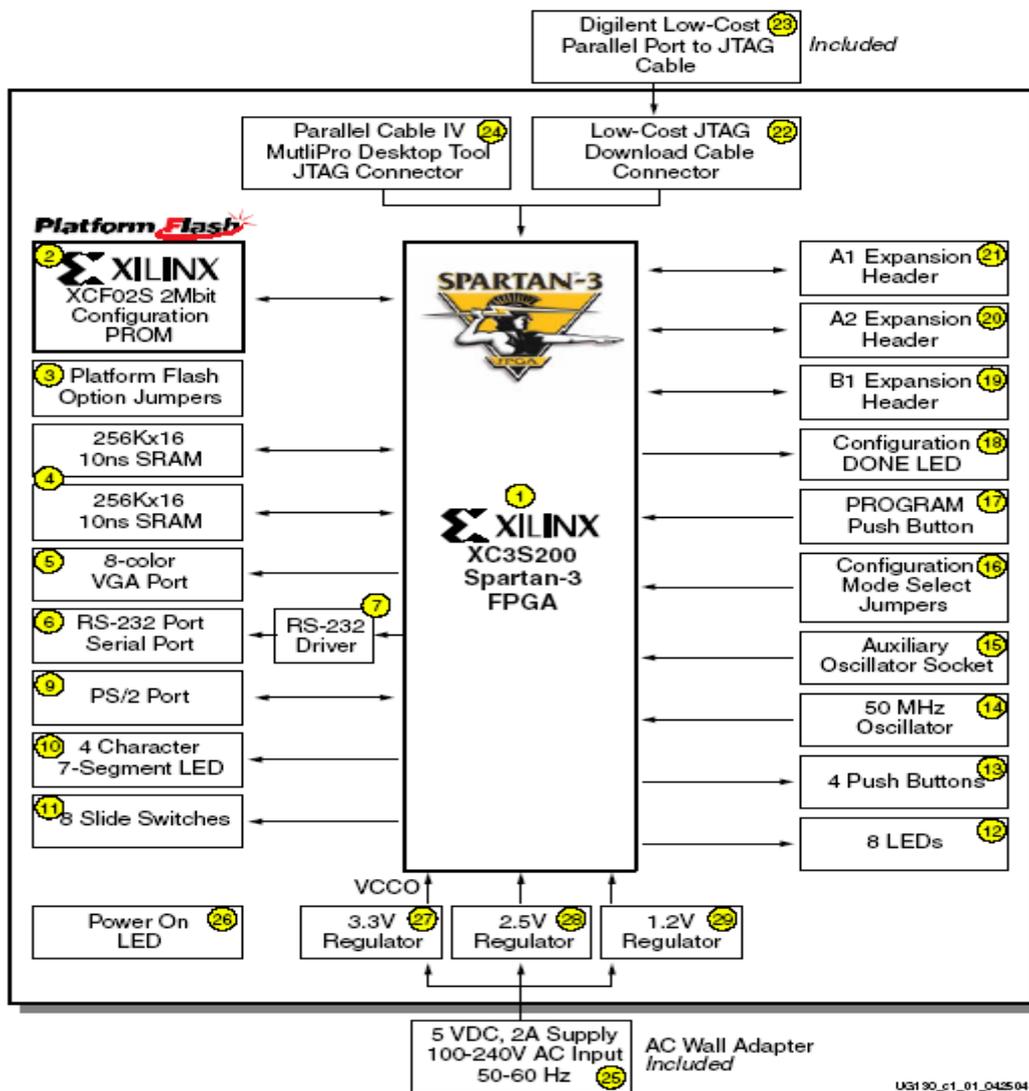


Figure: diagramme block de la Spartan 3

1. Circuit programmable de type FPGA (constructeur : Xilinx) avec une capacité d'intégration de 200 000 portes logiques. Le circuit FPGA est fourni suivant le package XC3S200FT256. (1)
2. Flash mémoire PROM d'une capacité de 2Mbit (2) :
 - 1Mbit de mémoire non volatile comme support de stockage de données et de programmes disponibles après la configuration du circuit FPGA.
 - Les options du cavalier permettent au FPGA de lire des données PROM ou une configuration du FPGA à partir d'autres sources (3) .

3. 1M byte de SRAM⁴.
 4. Port à 3 bits pour afficheur VGA à 8 couleurs⁵.
 5. Port série RS-232 à 9 pins⁶.
 6. Pilote pour le RS-232⁷.
 7. Port PS/2 pour souris/clavier⁸.
 8. Afficheur sept segments à quatre caractères¹⁰.
 9. huit swithes¹¹.
 10. Huit Leds individuelles¹².
 11. quatre boutons poussoirs¹³.
 12. Cristal oscillateur à 50 MHZ¹⁴.
 13. Socket pour une horloge supplémentaire¹⁵.
 14. Mode de configuration du FPGA sélectionné selon la position de cavaliers¹⁶.
 15. Bouton poussoir servant à forcer la reconfiguration du FPGA (une fois le bouton appuyé l'ancienne configuration est écrasée)¹⁷.
 16. Led indiquant que le FPGA a été correctement configurée¹⁸.
 17. 3 ports de connection à 40 pins chacun^{19 20 21}.
 18. Port JTAG²² pour câble de chargement²³.
 19. Câble JTAG Digilent pour chargement et debuggage connecté au port parallèle du PC²³.
 20. Port JTAG de chargement /debuggage compatible avec le câble parallèle de Xilinx²⁴.
 21. Entrée pour adaptateur AC²⁵.
 22. Led indicatrice (s'allume lors de la mise sous tension de la carte)²⁶.
- Régulateurs de tension 3.3 V²⁷, 2.5 V²⁸ et 1.2 V²⁹.

Bibliographie :

- [1]. Handbook of biometrics Anil K. Jain ; Patrick Flynn ; Arun A.Ross : Springer, 2005.
- [2]. Biometric Systems Technology, Design and Performance Evaluation ; James Wayman, Anil Jain, Davide Maltoni and Dario Maio(Eds) ; Springer 2004.
- [3]. Biometric Image Discrimination Technologies David Zhang ; Xiaoyuan Jing ; Jian Yang : IDEA GROUP PUBLISHING.2004.
- [4]. Champod Christophe : Reconnaissance automatique et analyse statistique des minuties sur l’empreinte digitale (thèse), Université de Lausanne. 2002.
- [5]. J-P. COCQUEREZ, « Analyse d’image : Filtrage et segmentation », Edition : MASSON, 1995.
- [6].A.J. Willis and L.Myers, « A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingerprints », *Pattern Recognition*, Vol. 34, pp. 255-270, 2001.
- [7]. [http : //www.visionindustrielle.org/visionwhat-biometrics.php](http://www.visionindustrielle.org/visionwhat-biometrics.php)
- [8]. L. Hong, Y. Wan, and Jain, « Fingerprint Image Enhancement : Algorithm and Performance Evaluation », *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no.8, pp.777-789,1998.
- [9]. N.Khang Pham, Détection de contours d’images, Rapport de Travaux d’intérêt personnel encadrés, l’institut de la francophonie pour l’informatique.2003.
- [10]. Galy Nicolas : Etude d’un système complet de reconnaissance d’empreintes digitales par un capteur microsysteme à balayage (thèse). 2003.
- [11]. R. Stefanelli, A. Rosenfld, some parallel thinning algorithms for digital pictures, *journal of the ACM*, vol. 18, N02, pp.255-264, April 1971.
- [12]. M. Yao « Quantifying Quality : A Case Study in Fingerprints », proceedings of IEEE Conference on AutoID, March 2002.
- [13]. Lin Hong. « Automatic Personal Identification Using Fingerprints », Ph.D.Thesis, 1998.
- [14]. « Evaluating Xilinx MicroBlaze for Network SoC solutions », *thèse de magister en the Engineering*, Peter Magnusson, 2004.
- [15]. <http://fvs.sourceforge.net>
- [16]. « EDK OS and libraries Reference Manual », *Embedded Development Kit 6.1i*. UG 114(V3.0), 2004.
- [17]. « MicroBlaze processor reference guide » ; *Embedded Development Kit 6.3it*. UG 081(V4.0), 2004.