

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO DE UN MODULO DE PROPIEDAD
INTELLECTUAL BASADO EN FPGA PARA
RECONOCIMIENTO DE HUELLAS DIGITALES”**

TESINA DE SEMINARIO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACION
ELECTRONICA Y AUTOMATIZACION INDUSTRIAL**

Presentado por:

Marco Antonio Jara Rivas

Guayaquil – Ecuador
2010

AGRADECIMIENTO

Agradezco a Dios por permitirme terminar con éxito el presente trabajo. A mi madre, padre, familiares que han estado presentes en los momentos que he necesitado apoyo.

DEDICATORIA

A Dios por darme salud, mis padres,
hermana, familiares y amigos que
siempre confiaron en mi.

TRIBUNAL DE SUSTENTACIÓN

Ing. Ronald Ponguillo
PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Sara Ríos
PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de Graduación de la ESPOL).

Marco Antonio Jara Rivas

RESUMEN

El proyecto consiste en la implementación de un sistema de identificación de dos huellas dactilares en un **FPGA** contenido en la tarjeta **DE2** de **ALTERA** con el objetivo de comparar las características de las huellas demostrando si son o no iguales.

El trabajo se realiza aplicando dos etapas. La primera, consiste en el uso de algoritmos en **MATLAB** para el mejoramiento de una imagen de huella dactilar escaneada, dando como resultado una huella binarizada y adelgazada en un formato de texto. La huella se plasma previamente en una hoja en blanco utilizando una almohadilla entintada diseñada para usos dactilográficos. En la segunda fase se desarrolla un algoritmo en **VHDL** para determinar el tipo y ubicación de las minucias (bifurcaciones y terminaciones) y posterior

comparación de acuerdo al porcentaje de coincidencias establecido previamente.

ÍNDICE GENERAL

1. SISTEMA BIOMETRICO PARA IDENTIFICACION PERSONAL	1
1.1 INDICADOR BIOMETRICO	1
1.2 CARACTERISTICAS DE UN SISTEMA BIOMETRICO	1
1.3 COMPONENTES DE UN SISTEMA BIOMETRICO	3
1.4 TIPOS DE SISTEMAS BIOMETRICOS	3
2. ANALISIS Y PROCESAMIENTO DE UNA HUELLA DACTILAR	7
2.1 DEFINICION Y CARACTERISTICAS DE UNA HUELLA DACTILAR	7
2.2 TRATAMIENTO DE LAS IMÁGENES	9
2.2.1 NORMALIZACION	10
2.2.2 SEGMENTACION	12
2.2.3 ORIENTACION DE CRESTAS	13
2.2.4 FRECUENCIA ESTIMADA DE CRESTAS	14

2.2.5 FILTRADO	16
2.2.6 BINARIZACION	17
2.2.7 ADELGAZAMIENTO	18
3. DETERMINACION Y COMPARACION DE MINUCIAS DE HUELLAS DACTLARES	19
3.1 EXTRACCION DE MINUCIAS	19
3.2 ALGORITMO DE CROSSING NUMBER	20
3.3 FALSAS MINUCIAS	21
3.4 COMPARACION DE HUELAS O MATCHING	23
4. RESULTADOS EXPERIMENTALES	26
4.1 TRATAMIENTO DE HUELLA DACTILAR EN MATLAB	26
4.2 IMPLEMENTACION DEL PROCESO DE EXTRACCION DE MINUCIAS Y COMPARACION EN TARJETA DE2 DE ALTERA	36
 CONCLUSIONES Y RECOMENDACIONES	
 ANEXOS	
 REFERENCIAS BIBLIOGRAFICAS	

ÍNDICE DE FIGURAS

FIGURA	DETALLE	PAG
1.1	BIOMETRIA FACIAL	4
1.2	HUELLA DACTILAR	4
1.3	GEOMETRIA DE LA MANO	5
1.4	IMAGEN DEL IRIS	5
1.5	IMAGEN DE UN FRAGMENTO DE VOZ	6
1.6	RECONOCIMIENTO POR FIRMA	6
2.1	UBICACIÓN DE CRESTAS Y VALLES	8
2.2	TIPOS DE MINUCIAS	9
2.3	MASCARA G _x	13
2.4	MASCARA G _y	13
2.5	IMAGEN DEL PROCEDIMIENTO PARA DETERMINAR FRECUENCIA DE CRESTAS	16
3.1	EJEMPLO DE UNA TERMINACION Y BIFURCACION	19
3.2	VENTANA 3x3 ANALISIS PIXEL P	20
3.3	MINUCIAS EN VENTANA DE 3x3	21

3.4	EJEMPLO DE FALSAS MINUCIAS	22
4.1	ALMOHADILLA ENTINTADA	27
4.2	HUELLA ESCANEADA DEL DEDO PULGAR DERECHO	27
4.3	HUELLA CON TAMAÑO STANDARD	28
4.4	IMAGEN DE HUELLA EN ESCALA DE GRISES	29
4.5	HUELLA NORMALIZADA	30
4.6	ORIENTACION DE CRESTAS	31
4.7	HUELLA CON FILTRO GABOR	32
4.8	HUELLA BINARIZADA	33
4.9	HUELLA ADELGAZADA	33
4.10	HUELLA ADELGAZADA CON TAMAÑO 216x216	34
4.11	HUELLA DACTILAR EN FORMATO DE TEXTO	35
4.12	PROCESO DE TRATAMIENTO DE HUELLAS	35
4.13	MEMORY INITIALIZATION FILE	37
4.14	MEGAWIZARD PLUG-IN MANAGER	38
4.15	MEGAWIZARD PLUG-IN MANAGER (PAGE 2)	39
4.16	MEGAWIZARD PLUG-IN MANAGER:ROM 1 PORT	40
4.17	MEGAWIZARD PLUG-IN MANAGER:ROM 1 PORT (PAGE2)	41
4.18	MEGAWIZARD PLUG-IN MANAGER:ROM 1 PORT (PAGE3)	42
4.19	MEGAWIZARD PLUG-IN MANAGER:ROM 1 PORT (SUMARY)	42

4.20	PROCESO DE IMPLEMENTACION EN TARJETA DE2	43
4.21	PROCESO DE EXTRACCION DE MINUCIAS	45
4.22	PROCESO DE COMPARACION DE MINUCIAS ENTRE 2 HUELLAS	48
4.23	HUELLA1_1 RESOLUCION 500 DPI	52
4.24	HUELLA1_1 CON MINUCIAS RESALTADAS	52
4.25	HUELLA1_2 RESOLUCION 500 DPI	53
4.26	HUELLA1_2 CON MINUCIAS RESALTADAS	53

ÍNDICE DE ECUACIONES

ECUACION	DETALLE	PAG
2.1	MEDIA DE UNA IMAGEN $N \times N$	11
2.2	VARIANZA DE UNA IMAGEN $N \times N$	11
2.3	NORMALIZACION DE UNA IMAGEN $N \times N$	11
2.4	VARIANZA DE UN BLOQUE $W \times W$	12
2.5	ORIENTACION DEL BLOQUE $W \times W$	14
2.6	COVARIANZA EN X DEL BLOQUE $W \times W$	14
2.7	COVARIANZA EN Y DEL BLOQUE $W \times W$	14
2.8	FRECUENCIA DE CRESTAS DEL BLOQUE $W \times W$	15
2.9	FILTRO GABOR	16
3.1	CROSSING NUMBER	20
4.1	COORDENADAS DEPENDIENTES DE DIRECCION DE MEMORIA	46

ÍNDICE DE TABLAS

TABLA	DETALLE	PAG
3.1	RELACION CROSSING NUMBER CON TIPO DE MINUCIA	20
4.1	CANTIDAD DE MINUCIAS, RESULTADOS DEL ALGORITMO DE EXTRACCION	50
4.2	CANTIDAD DE MINUCIAS, RESULTADOS VISUALES	50
4.3	EFICIENCIA EN LA EXTRACCION DE MINUCIAS	51
4.4	RESULTADOS DE COMPARACION DE DOS HUELLAS	54

INTRODUCCIÓN

La biometría es una tecnología que se ha venido estudiando desde tiempo atrás y es considerada en la actualidad como el método más aceptado de identificación humana, basada en el reconocimiento de una de una característica física e intransferible de las personas.

Los sistemas biométricos incluyen un dispositivo de captura y un software; el primero obtiene la información del medio físico y la transforma en un formato digital para que el segundo analice los datos y emita un resultado. El sistema biométrico más utilizado es el de huella digital donde sus aplicaciones abarcan un gran número de sectores: desde el acceso seguro a computadores hasta el control de acceso físico a una área restringida.

Una huella digital es única para cada persona y no cambia a través del tiempo, la misma que está formada por una serie de valles y crestas las cuales forman estructuras específicas llamadas minucias (resaltan las dos más importantes las **Terminaciones** y **Bifurcaciones**). Cada minucia posee una característica y

posición única en la huella que puede ser medida posibilitando la elaboración de la identidad de una persona a partir de estas.

En este proyecto se muestra el procesamiento de una huella dactilar en Matlab y el desarrollo de algoritmos en VHDL. Uno para determinar el tipo minucia (bifurcación o terminación) y otro para la comparación de las características de dos huellas. Se realizan pruebas con huellas auténticas verificando el porcentaje de eficiencia de los algoritmos dependiendo de la calidad de las imágenes de las huellas procesadas.

Capítulo 1

1. SISTEMA BIOMETRICO PARA IDENTIFICACION PERSONAL

1.1 INDICADOR BIOMETRICO

Un indicador biométrico es una característica que puede ser medible y utilizable para propósitos de identificación de personas, el mismo que debe cumplir los siguientes requerimientos:

1. **Universalidad:** cualquier persona posee esa característica.
2. **Unicidad:** baja probabilidad de la existencia de dos individuos con una misma característica.
3. **Permanencia:** la característica no cambia en el tiempo.
4. **Cuantificación:** la característica puede ser medida en forma numérica.

1.2 CARACTERISTICAS DE UN SISTEMA BIOMETRICO

Un sistema biométrico con utilidad práctica necesita desempeñarse con exactitud y rapidez para que el grado de aceptabilidad y confiabilidad en las personas sea el ideal y poderse utilizar de manera segura sin causar daño a usuario alguno.

Un sistema biométrico debe generar confianza ya que se puede generar un impacto psicológico que altere el uso rutinario del sistema, como por ejemplo reconocimiento por retina, en donde una persona expone su ojo ante un dispositivo que emite un haz para poder obtener patrones de reconocimiento.

Otra característica importante es la robustez que debe tener el sistema, al no ser fácil de eludir e identificar a el usuario correcto en un porcentaje mayor al 98%.El sistema debe ser capaz de reconocer una característica de un ser muerto ya que es una manera de burlar el sistema. Actualmente un sistema biométrico tiene todas las garantías para el usuario pues no solo se limita a identificar las características sino a detectar otros elementos innatos de un ser humano como por ejemplo un sistema basado en el reconocimiento del iris revisa patrones característicos en las manchas de éste, un sistema infrarrojo para chequear las venas de la mano detecta flujos de sangre caliente y lectores de ultrasonido para huellas dactilares revisan estructuras subcutáneas de los dedos.

1.3 COMPONENTES DE UN SISTEMA BIOMETRICO

Los dispositivos biométricos poseen tres componentes básicos, el primero se encarga de la adquisición análoga o digital de algún indicador biométrico de una persona, el segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos con los almacenados y el tercero establece una interfaz con aplicaciones diversas amigables para el usuario.

Un sistema biométrico realiza dos funciones principales que son la adquisición de datos de un indicador perteneciente a un individuo que se la realiza por medio de un lector biométrico para la extracción de características útiles y la identificación o reconocimiento que procesa las características y compara con una base de datos estableciendo la identidad de una persona.

1.4 TIPOS DE SISTEMAS BIOMETRICOS

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características. Las técnicas biométricas más conocidas están basadas en los siguientes indicadores:

1.4.1 ROSTRO

Las técnicas de reconocimiento basadas en los rasgos del rostro, conocidas como biometría facial

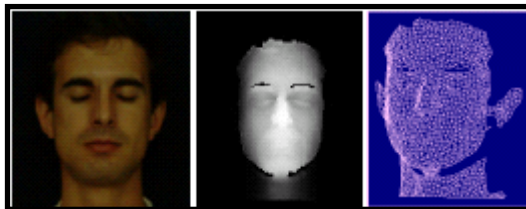


FIGURA 1.1. BIOMETRIA FACIAL

1.4.2 HUELLAS DACTILARES

Es un rasgo particular de cada individuo cuyo origen tiene lugar durante la etapa fetal permaneciendo inmutable durante toda la vida.



FIGURA 1.2. HUELLA DACTILAR

1.4.3 GEOMETRIA DE LA MANO

Es un rasgo que no proporciona muy altos niveles de fiabilidad, si bien es percibido como poco intrusivo por los usuarios, lo que le dota de una alta aceptación como medio de autenticación.



FIGURA 1.3. GEOMETRIA DE LA MANO

1.4.4 IRIS

El iris, como herramienta de autenticación, tiene ventajas frente a otras partes del cuerpo, pues no se modifica en el transcurso de la vida y está protegido por la córnea, que lo hace inmune a la influencia del ambiente.



FIGURA 1.4. IMAGEN DE IRIS

1.4.5 VOZ

Depende de las características de la estructura física del tracto vocal de un individuo así como también de sus características de comportamiento.

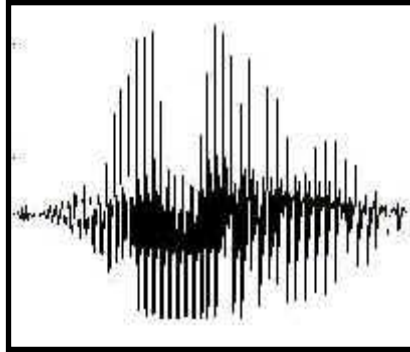


FIGURA 1.5. IMAGEN DE UN FRAGMENTO DE VOZ

1.4.6 FIRMA

La dificultad tecnológica del reconocimiento a través de la firma estriba en saber diferenciar los rasgos que nunca cambian, y los que sí lo hacen en cada firma.



Capítulo 2

2. ANALISIS Y PROCESAMIENTO DE UNA HUELLA DACTILAR

2.1 DEFINICION Y CARACTERISTICAS DE UNA HUELLA DACTILAR

Una huella dactilar es una impresión de los relieves epidérmicos contenidos en la yema de los dedos producido al contacto con una superficie lisa o porosa. Cada huella esta compuesta por una serie de Crestas y Valles que disponen diferentes formas en la epidermis de los dedos siendo una característica única en cada individuo, originándose en la etapa fetal y permaneciendo inmutable a lo largo de toda la vida.

A lo largo de los años muchos investigadores han aportado grandes estudios científicos acerca de la estructura de las huellas digitales, definiendo a las crestas como la parte que sobresale en la epidermis de la yema de los dedos. A principios del siglo pasado se llegaron a varias conclusiones biológicas acerca de las huellas dactilares:

- La estructura de las crestas y valles de la epidermis de un individuo es única.
- La estructura de las crestas y valles de un individuo puede variar pero lo hace dentro de límites muy reducidos.
- Los detalles de las estructuras de crestas y valles son particulares de cada individuo e invariables en el tiempo.

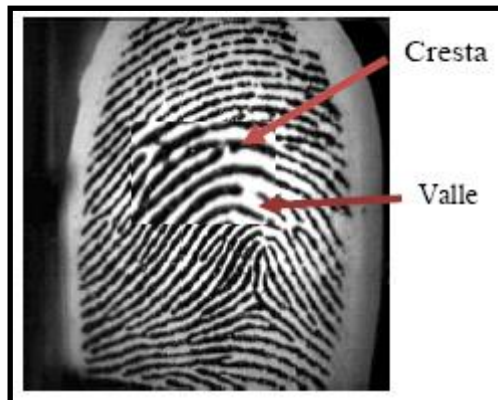


FIGURA 2.1. UBICACIÓN DE CRESTAS Y VALLES

Las crestas forman estructuras específicas llamadas minucias que son las características medibles apropiadas para desarrollar algoritmos de identificación de huellas. Dentro de la gama de formas que las minucias pueden tomar en una huella, existe una clasificación de figuras básicas que

se ha tomado como objeto de estudio las cuales se muestran en la siguiente figura.

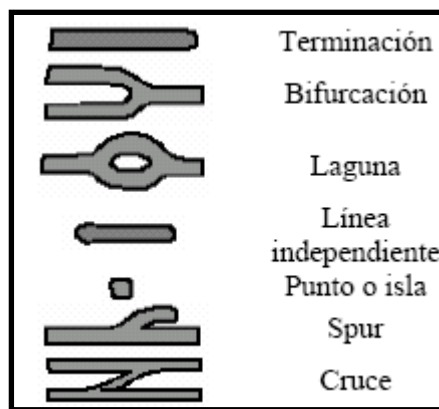


FIGURA 2.1. TIPOS DE MINUCIAS

Las minucias utilizadas en el proyecto son las Terminaciones y Bifurcaciones por la facilidad de determinarlas de acuerdo a algoritmos existentes tales como Crossing Number.

2.2 TRATAMIENTO DE LAS IMÁGENES

Es de vital importancia establecer un tamaño definido de las imágenes de las huellas a tratar. En este proyecto se captura la imagen escaneando la huella dactilar, obteniendo un archivo JPGE el mismo que se lo utiliza para ser

procesado en **MATLAB**, en donde se fija el tamaño de 216x 216 Píxeles para la imagen resultante.

Los algoritmos que se implementaron en MATLAB para el tratamiento de las imágenes son los siguientes y se ejecutan en el orden establecido:

- 1) Normalización**
- 2) Segmentación**
- 3) Orientación de crestas**
- 4) Frecuencia estimativa de crestas**
- 5) Filtrado**
- 6) Binarización**
- 7) Adelgazamiento**

2.2.1 NORMALIZACION

La etapa de Normalización tiene como objetivo resaltar las crestas en una imagen de huella dactilar en escala de grises. Las intensidades en escala de grises varían de 0 a 255 siendo el valor 0 el elemento mas oscuro y 255 el mas claro y estas dependen de la claridad y calidad en que se captura una

imagen. En esta etapa se determina la media y la varianza de las intensidades de los píxeles de la imagen por medio de las siguientes ecuaciones:

$$M(I) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j)$$

ECUACION 2.1. MEDIA DE UNA IMAGEN N x N

$$VAR(I) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I(i, j) - M(I))^2$$

ECUACION 2.2. VARIANZA DE UNA IMAGEN N x N

Donde $M(I)$ es la media de la intensidad de los píxeles de una imagen, $I(i,j)$ representa a la intensidad los píxeles en las coordenadas (i,j) , N es el tamaño de la imagen asumiendo que es de $N \times N$, $VAR(I)$ es la varianza en la intensidad de los píxeles.

Al obtener la media (M) y la varianza (V) de imagen se procede a unificar la intensidad de cada píxel partiendo de una media (M_0) y varianza (V_0) esperadas, de acuerdo con la siguiente ecuación:

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & \text{Si } I(i, j) > M, \\ M_0 - \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & \text{otros} \end{cases},$$

ECUACION 2.3. NORMALIZACION DE UNA IMAGEN N x N

2.2.2 SEGMENTACION

Después de la etapa de Normalización empieza la etapa de Segmentación que tiene como objetivo separar la información relevante la cual contiene las minucias del fondo de la imagen. Una imagen de huella dactilar posee la zona que contiene la información a utilizar y el fondo. La primera tiene una varianza significativa en la intensidad de sus píxeles ya que esta compuesta por las crestas y valles que generalmente tienen valores de intensidad cercanos a los extremos opuestos y la segunda posee mas homogeneidad por lo que la varianza es insignificante.

El algoritmo de segmentación divide la imagen en bloques de un tamaño de 16x16 y calcula la varianza, la misma que si esta por debajo de un valor de umbral preestablecido se considera que es el fondo asignando un valor de intensidad a todo el bloque y si esta por encima del umbral se copia la imagen normalizada. La siguiente formula se aplica para el cálculo de la varianza por bloques:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (I(i, j) - M(k))^2$$

ECUACION 2.4. VARIANZA DE UN BLOQUE WxW

Donde $V(k)$ es la varianza del bloque k , $I(i,j)$ es el valor de la intensidad de los píxeles en las coordenadas (i,j) de la imagen normalizada, $M(k)$ es la media del valor de la intensidad de los píxeles en el bloque k y W es el tamaño del bloque $W \times W$.

2.2.3 ORIENTACION DE CRESTAS

Después de la etapa de normalización y segmentación viene la etapa de orientación de crestas que tiene como objetivo calcular la alineación o dirección de cada píxel que conforma las crestas de la imagen normalizada para en posteriores etapas filtrar la imagen de acuerdo a la orientación de los píxeles.

Para calcular de la orientación se debe seguir los siguientes pasos:

- Se divide la imagen normalizada en bloques de $W \times W$ (generalmente se utiliza 16×16) y se calcula la magnitud de los gradientes en X y Y ($\partial_x(i, j)$ y $\partial_y(i, j)$) para cada píxel del bloque utilizando los siguientes operadores Sobel para cada coordenada:

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

FIGURA 2.3. MASCARA Gx

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

FIGURA 2.4. MASCARA Gy

- Luego de calcular el gradiente en X y Y de cada píxel se procede a calcular la orientación del bloque de acuerdo a su píxel central con las siguientes ecuaciones:

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \frac{V_y(i, j)}{V_x(i, j)},$$

ECUACION 2.5. ORIENTACION DEL BLOQUE WxW

$$V_x(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2\partial_x(u, v)\partial_y(u, v),$$

ECUACION 2.6. COVARIANZA EN X DEL BLOQUE WxW

$$V_y(i, j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \partial_x^2(u, v)\partial_y^2(u, v),$$

ECUACION 2.7. COVARIANZA EN Y DEL BLOQUE WxW

2.2.4 FRECUENCIA ESTIMATIVA DE CRESTAS

Luego de obtener la orientación se procede a estimar la frecuencia de las crestas ya que es un parámetro fundamental para el filtrado y mejoramiento de la imagen. La frecuencia es en si la cantidad de crestas por unidad de longitud a lo largo de un eje de coordenadas (x,y), en dirección ortogonal a la orientación de las crestas partiendo de un punto central perteneciente a un

bloque de tamaño $W \times W$ de una imagen normalizada. Para determinar la frecuencia de las crestas se debe seguir los siguientes pasos:

- Se divide la imagen normalizada en bloques de tamaño 32×32 píxeles.
- La orientación de las crestas da la dirección del plano en el que se determinara la frecuencia del bloque considerando el eje X perpendicular y el eje Y paralelo a la orientación de las crestas.
- A lo largo del eje X se comienza a sumar la cantidad de picos de intensidad de píxeles (correspondiente a las crestas) del bloque de la imagen normalizada, siendo este valor la cantidad de crestas en el bloque. El valor de longitud total es la suma de las distancias de cresta a cresta en cada bloque. Los dos valores se relacionan inversamente mediante la formula:

$$FRECUENCIA = \frac{\#CRESTAS}{LONGUITUD}$$

ECUACION 2.8. FRECUENCIA DE CRESTAS DEL BLOQUE $W \times W$

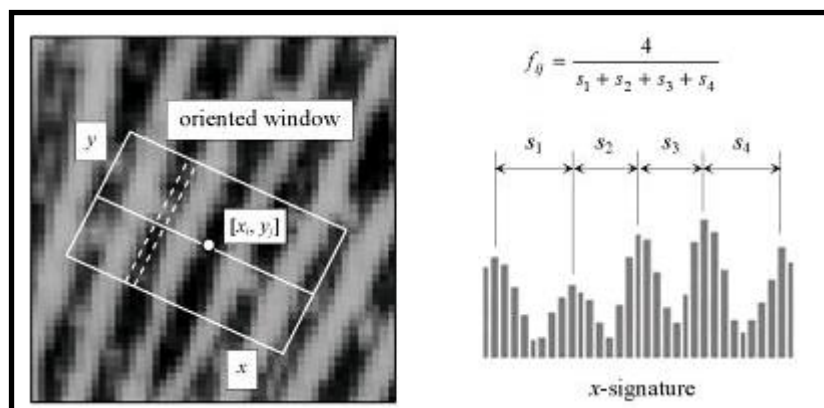


FIGURA 2.5. IMAGEN DEL PROCEDIMIENTO PARA DETERMINAR FRECUENCIA DE CRESTAS.

2.2.5 FILTRADO

Después de obtener la orientación y la frecuencia de las crestas se procede a ejecutar la etapa de filtrado de la huella utilizando el filtro de detección de bordes **GABOR** el cual tiene una característica particular que es la selectividad en frecuencia y orientación. Las imágenes se las filtran en ángulos y frecuencias específicas eliminando el ruido y realizando de una manera limpia la estructura de crestas y valles logrando una diferencia muy evidente entre ellas.

El filtro **GABOR** se lo representa mediante la siguiente expresión

$$G(X, Y) = \text{Cos}(2\pi * F * X_{\theta}) * e^{-0.5 \left(\frac{X_{\theta}^2}{\sigma_X^2} + \frac{Y_{\theta}^2}{\sigma_Y^2} \right)}$$

ECUACION 2.9. FILTRO GABOR

Donde,

$$X_{\theta} = X\cos\theta + Y\sin\theta$$

$$Y_{\theta} = Y\cos\theta - X\sin\theta$$

θ = Orientación de cresta

F = Frecuencia de cresta

$$\sigma_x = 2/F$$

$$\sigma_y = 2/F$$

El filtro se aplica a todos los píxeles en escala de grises que forman parte de la huella (no segmentados) de acuerdo a su orientación y frecuencia.

2.2.6 BINARIZACIÓN

La binarización es una técnica en el que convierte una imagen de color o escala de grises a una en blanco y negro. Después de haber obtenido la imagen de la huella filtrada se procede a pasar el valor de intensidad de los píxeles no segmentados de escala de grises a binario, fijando a 1 el valor de las crestas y 0 el de los valles teniendo en cuenta que en la imagen binaria se mostrará las crestas de color negro y los valles de color blanco. Para binarizar se toma un valor base en escala de grises estableciendo blanco o negro para un píxel analizado si el valor de este en escala de grises esta por encima o por debajo del valor base.

2.2.7 ADELGAZAMIENTO

Después de haber binarizado la imagen se ejecuta la etapa de adelgazamiento o esqueletización que consiste en erosionar sucesivamente el grosor de las crestas de la imagen binarizada hasta que llegue al grosor de un píxel. Es importante este paso porque permite utilizar el algoritmo de crossing number para determinar los tipos de minucias que tiene la huella. En el proyecto se utilizó el comando de MATLAB “**thin**” para adelgazar la imagen siendo el método más rápido al no elaborar un algoritmo propio para el adelgazamiento.

Capítulo 3

3. DETERMINACION Y COMPARACION DE MINUCIAS DE HUELLAS DACTILARES

3.1 EXTRACCION DE MINUCIAS

La clave para la identificación de huellas es determinar que tipo de minucias existen y que posición o coordenadas tienen en la imagen. Al tener una imagen binarizada y adelgazada de una huella dactilar se procede a extraer los puntos característicos, tales como bifurcaciones y terminaciones, que son los que se tomarán en cuenta para el análisis de la huella. Para saber que tipo de minucia existe en una huella se analiza cada píxel que forme parte de las crestas aplicando el algoritmo de Crossing Number.



(a) TERMINACION



(b) BIFURCACION

FIGURA3.1. EJEMPLO DE UNA TERMINACION Y BIFURCACION

3.2 ALGORITMO DE CROSSING NUMBER

El algoritmo de Crossing Number analiza los vecinos de cada píxel que forma parte de las crestas de la huella mediante una ventana de 3 x 3.

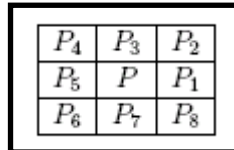


FIGURA 3.2. VENTANA 3X3 ANALISIS PIXEL P

Se realiza operaciones algebraicas con la vecindad del píxel en análisis mediante la Formula de Crossing Number.

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}|, \quad P_9 = P_1$$

ECUACION 3.1. CROSSING NUMBER

El resultado de aplicar Crossing Number determina un tipo de minucia que se lo aprecia en la siguiente tabla:

CROSSING NUMBER	MINUCIA
0	PUNTO AISLADO
1	TERMINACION
2	CONTINUACION
3	BIFURCACION

TABLA 3.1. RELACION CROSSING NUMBER CON TIPO DE MINUCIA

Se puede apreciar las minucias en una ventana de 3x3 en la siguiente figura:



FIGURA3.3. MINUCIAS EN VENTANA DE 3X3

3.3 FALSAS MINUCIAS

Después de aplicar el algoritmo de adelgazamiento a la imagen de una huella binarizada generalmente se generan falsas características que el algoritmo de Crossing Number detecta como minucias validas, por lo que se recomienda hacer un paso previo de identificación y eliminación de las mismas. A continuación ejemplos más comunes de minucias falsas:

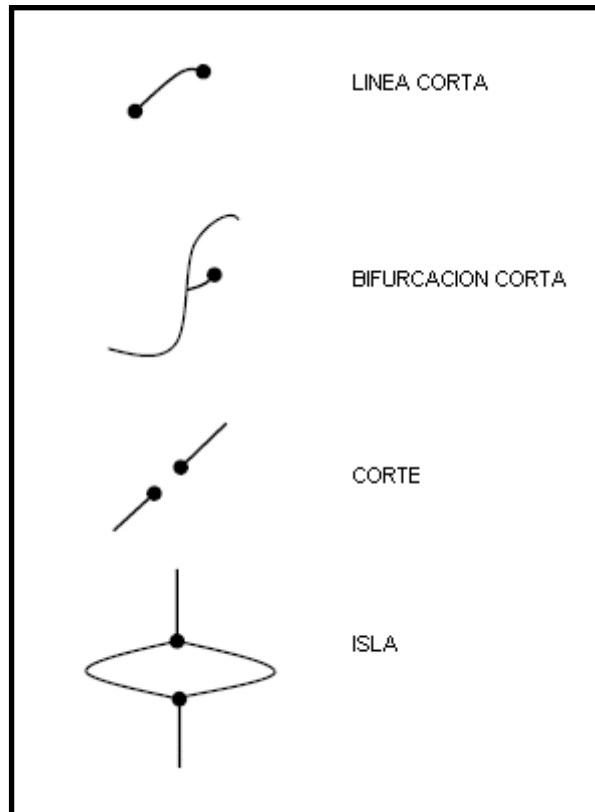


FIGURA3.4. EJEMPLO DE FALSAS MINUCIAS

Línea Corta, Si la distancia entre dos Terminaciones que forma parte de una misma línea continua es menor a un valor predeterminado en píxeles.

Bifurcación Corta, Si la distancia entre una Terminación y una Bifurcación es menor a un valor predeterminado en píxeles.

Corte, Si la distancia entre dos Terminaciones que no forma parte de una misma línea continua es menor a un valor predeterminado en píxeles.

Isla, Si la distancia entre dos Bifurcaciones que forman parte de una misma línea continua es menor a un valor predeterminado en píxeles.

3.4 COMPARACION DE HUELLAS O MATCHING

Después de extraer las características, la huella dactilar posee su identificación mediante la cantidad y tipo de minucias (de las cuales se tomaron en cuenta Bifurcaciones y Terminaciones) así como sus coordenadas en la imagen. Esta información se convierte en vectores específicos de la huella, la misma que se compara con otra de acuerdo a un porcentaje de coincidencias predeterminado, concluyendo si son o no huellas idénticas.

Después de obtener la información de las características que tienen las huellas a compararse, que en este caso sería una cantidad X de Bifurcaciones y una cantidad Y de Terminaciones, cada una con su coordenada correspondiente en la imagen, se procede a comparar cada minucia de una huella (en tipo y coordenada) con otra obteniendo una cantidad de coincidencias. Se establece previamente un porcentaje de aceptación de acuerdo a la cantidad de minucias que posee las huellas a comparar por lo que si las coincidencias igualan o superan el porcentaje de

aceptación entonces las huellas comparadas son iguales y si es inferior sería lo contrario. Cabe recalcar que este método solo compara el tipo de minucia y las coordenadas en la imagen por lo que se considera un método efectivo, fácil de realizar y con resultados satisfactorios.

En varios textos sugieren agregar características adicionales a las minucias extraídas de las huellas para tener un resultado más preciso en la comparación de las huellas. Se sugiere no solo tener el tipo de minucia (bifurcación o terminación) y las coordenadas de estas en la imagen como información de una huella para su comparación, sino también tomar distancias desde una minucia referencial hacia sus vecinas obteniendo así un parámetro adicional para utilizarlo en la comparación. Este procedimiento se lo realiza a cada minucia y las distancias que se toman se las hacen entre minucias del mismo tipo.

Se puede determinar la eficiencia del sistema de acuerdo a los resultados experimentales determinando los siguientes parámetros:

FALSAS ACEPTACIONES (F.A): Es cuando dos huellas son diferentes y el sistema concluye que son idénticas.

FALSOS RECHAZOS(F.R): Ocurre cuando dos huellas son iguales y el sistema determina que son diferentes.

% DE ERROR: $((F.R+F.A)*100) / \text{TOTAL COMPARACIONES HECHAS}$

EFICIENCIA: $100 - \% \text{ DE ERROR}$

Capítulo 4

4. RESULTADOS EXPERIMENTALES

4.1 TRATAMIENTO DE HUELLA DACTILAR EN MATLAB

El objetivo es usar el programa Matlab para mejorar, binarizar y adelgazar la imagen de una huella dactilar. Se utilizaron algoritmos para el mejoramiento de las imágenes y para la conversión de un archivo de Matlab a uno de texto con el propósito de facilitar la transferencia de la información de la huella a la Tarjeta DE2 de Altera.

Se describe a continuación los resultados del tratamiento de la huella dactilar y procesos a seguir:

Se decidió plasmar la huella dactilar en una hoja en blanco usando una almohadilla entintada especial utilizada exclusivamente en dactilografía.



FIGURA4.1. ALMOHADILLA ENTINTADA

Luego de tener impresa la huella dactilar en una hoja se procede a escanearla a una resolución de 500 DPI a un tamaño promedio de 330 x 500 Píxeles para el dedo pulgar. Cabe recalcar que a una resolución menor el tamaño de la imagen disminuye.



FIGURA4.2. HUELLA ESCANEADA DEL DEDO PULGAR DERECHO

Al ser escaneada la huella se la graba como un archivo JPEG para ser utilizada en MATLAB. Se procede a aplicar un algoritmo previo en donde se le cambia el tamaño de la imagen a uno Standard de 256x296 Píxeles que se aplicara a todas las imágenes sin excepción.



FIGURA4.3. HUELLA CON TAMAÑO STANDARD

Ahora se procede a pasar la imagen a escala de grises ya que los procesos para mejorar la imagen se los realizan en este tipo de tono.



FIGURA4.4. IMAGEN DE HUELLA EN ESCALA DE GRISES

Después de pasar la imagen a escala de grises se procede a normalizarla y segmentarla. En esta etapa se resaltan las crestas y se diferencia la parte en donde se encuentra la información de la huella con la que es irrelevante. En el proceso de segmentación se utilizo bloques de 16 x 16 píxeles, ya que es la dimensión recomendada para realizar en esta etapa y en el proceso de normalización se trabajo con valores de intensidad de píxeles en un rango de 0 a 1 donde 0 es negro y 1 blanco. Se usó como media esperada el valor de 1 y como varianza esperada el valor de 0 lo que dio un resultado como se ilustra en la figura 4.5.



FIGURA4.5. HUELLA NORMALIZADA

Luego se procede a determinar la orientación de las crestas, donde se utilizaron bloques de 16x16 píxeles para analizar la imagen y obtener un óptimo resultado. El resultado es una matriz con valores en radianes de la orientación que tiene cada píxel en la imagen.



FIGURA4.6. ORIENTACION DE CRESTAS

Inmediatamente después de obtener la orientación de las crestas se determina la frecuencia de las crestas que es el parámetro faltante para aplicar el filtro Gabor. Se definió bloques de 32x32 para determinar las frecuencias estimativas, barriendo toda la imagen a fin de obtener una matriz de frecuencias de la imagen. Ya teniendo la orientación y frecuencia de crestas para cada píxel, entonces se procede a filtrar la imagen de la huella segmentada (información relevante) mediante el filtro Gabor cuya principal característica es la exclusividad en frecuencia y orientación para cada píxel. Se inicializo $\sigma_x = 0.7$ y $\sigma_y = 0.7$ dando el mejor resultado en filtraje de la

imagen. Después del filtraje se procede a binarizar la imagen definiendo un valor umbral y luego se adelgaza mediante la operación morfológica “**bwmorph**” con característica “**thin**” e “**inf**” los cuales indican adelgazar hasta el infinito o hasta que ya no se pueda. Luego se secciona la imagen para que quede de tamaño 216x216 píxeles que es el tamaño referencial de los datos que se van a procesar en la tarjeta DE2.



FIGURA4.7. HUELLA CON FILTRO GABOR



FIGURA4.8. HUELLA BINARIZADA

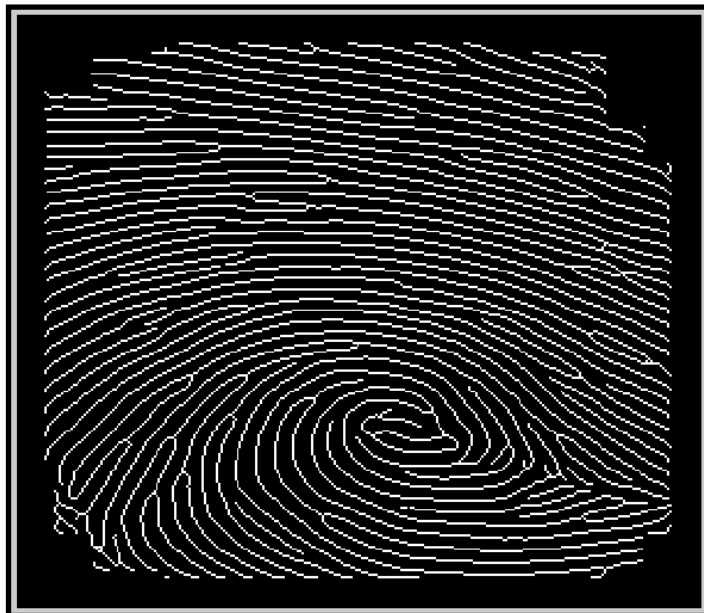


FIGURA4.9. HUELLA ADELGAZADA

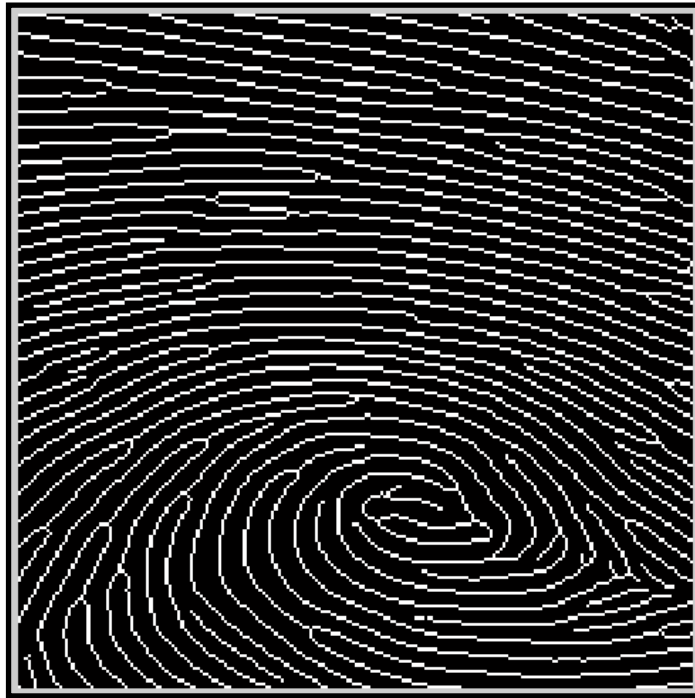


FIGURA4.10. HUELLA ADELGAZADA DE TAMAÑO 216X216

El paso final es pasar la imagen de la huella adelgazada de tamaño 216x216 píxeles a un formato de texto o .txt que nos servirá como datos que posteriormente se ubicara en una ROM en el FPGA de la tarjeta DE2. El archivo de texto tendría un aspecto como se muestra en la figura 4.11.

En la figura 4.12 se describe el diagrama de bloques del proceso realizado en Matlab, partiendo de una huella escaneada a 500 DPI la misma que se procesa filtrando, binarizando y adelgazando para que se aplique un algoritmo que lleve la huella resultante a formato de texto y ser la información de una memoria ROM contenida en el FPGA de la tarjeta DE2.

4.2 IMPLEMENTACION DEL PROCESO DE EXTRACCION DE MINUCIAS Y COMPARACION EN TARJETA DE2 DE ALTERA

Para este proyecto se utiliza la tarjeta DE2 de Altera que contiene un FPGA Cyclone II 2C35 con 672 pines en donde se va a implementar los algoritmos desarrollados en VHDL, para la extracción de minucias y comparación de dos huellas dactilares. Se utilizará el programa Quartus II versión 9.1 como plataforma de creación de los algoritmos y medio de comunicación con el FPGA de la tarjeta DE2.

Después del procesamiento de una huella dactilar en Matlab queda la información en un archivo de texto el mismo que será ingresado a la tarjeta DE2 en una memoria ROM interna del FPGA en la que los datos se encontraran almacenados en un Memory Initialization File que es un archivo de memoria con extensión **.mif** exclusivo de **ALTERA** que se lo crea siguiendo estos pasos:

- 1) Al abrir el Quartus II se ubica en la pestaña **FILE**.
- 2) Se situa en **NEW**.
- 3) Se elige **MEMORY INITIALITATION FILE** en donde se especifica el número de palabras y el tamaño de ellas.

En nuestro caso la cantidad de palabras seria 46656 (ya que la matriz de datos es de 216 x 216 lo que equivale a 46656 elementos) cada una de 1 bit de tamaño. Luego se procede a copiar los datos que se obtuvieron después del procesamiento de la huella en MATLAB en el Memory Initialization File quedando como se indica en la figura 4.13.

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	1	1	1	1	1	1	1	1
8	1	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	1	1	1	1	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
64	0	0	0	0	0	1	1	1
72	1	1	1	1	0	0	0	0
80	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0
104	0	0	0	0	0	0	0	0
112	0	1	1	1	1	1	0	0
120	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0
136	0	0	0	0	0	1	1	1
144	1	0	0	0	0	0	0	0
152	0	0	0	0	0	1	1	0
160	0	0	0	0	0	0	0	0
168	0	0	0	0	0	0	0	0
176	1	1	1	0	0	0	0	0

FIGURA4.13. MEMORY INITIALITATION FILE

Ahora se procede a crear una ROM que es exclusiva para plataformas de Altera mediante los pasos siguientes:

- 1) Se ubica en la pestaña **TOOLS**
- 2) Se selecciona **MEGAWIZARD PLUG-IN MANAGER**

3) Se elige **Create a New custom Megafuntion variation** como se ve en la figura 4.14.

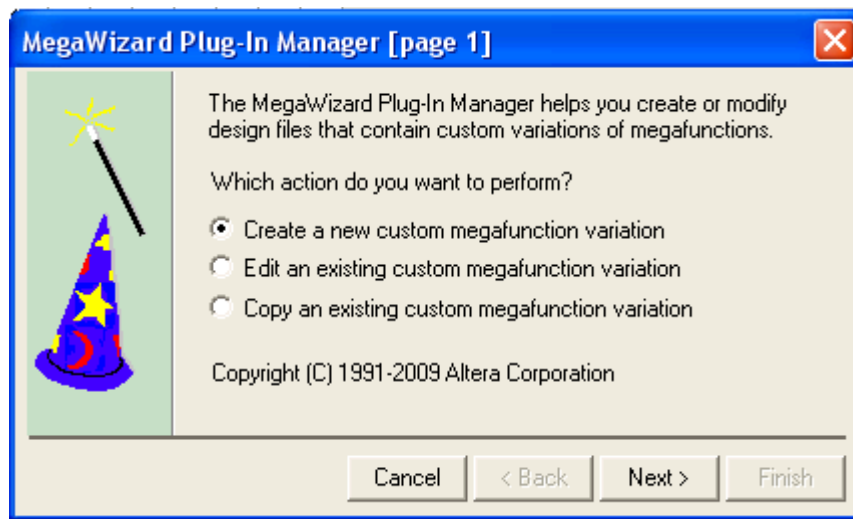


FIGURA4.14. MEGAWIZARD PLUG-IN MANAGER

Luego de pulsar **Next** se llega a otra pantalla donde se elige el tipo de megafuntion el cual se eligió **ROM: 1PORT**, Familia a usarse **CYCLONE II**, tipo de archivo **VHDL** , el nombre del archivo, tal como se ve en la figura 4.15.

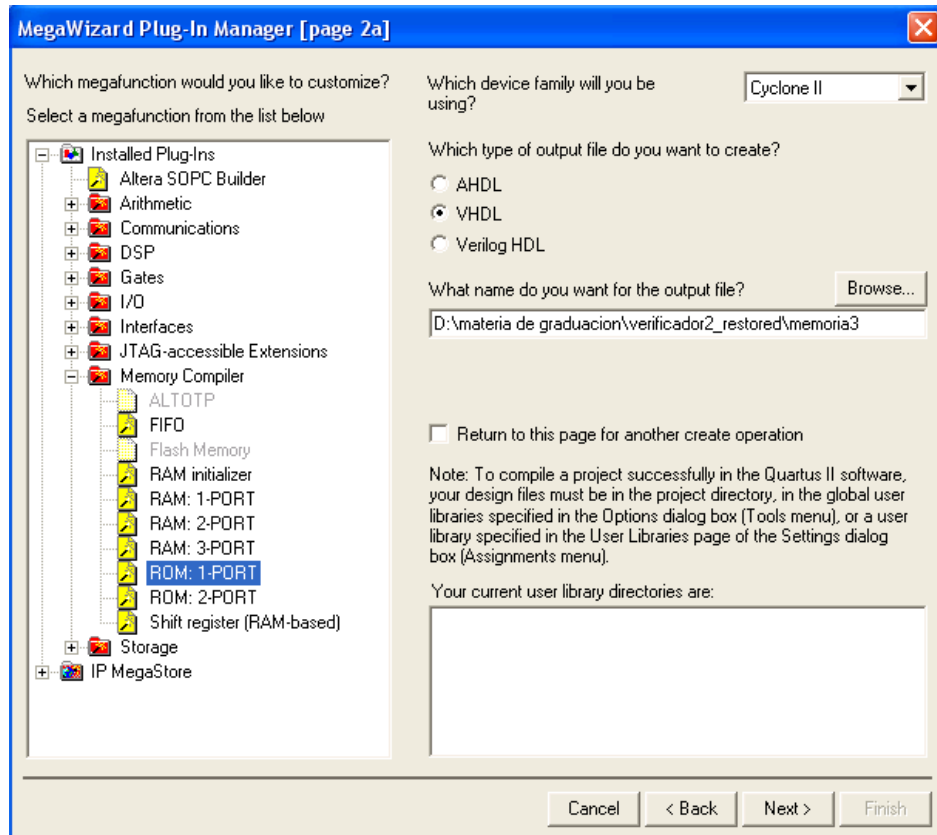


FIGURA4.15. MEGAWIZARD PLUG-IN MANAGER (PAGE 2)

En la siguiente pantalla se elige las características de la ROM que se va a crear tal como el bus de salida que en este caso va a ser **1**, el número de palabras en la memoria que va a ser la máxima **65536** ya que el archivo MIF tiene 46656 elementos, así mismo se elige un solo Clock para el sistema, tal como se muestra en la figura 4.16.

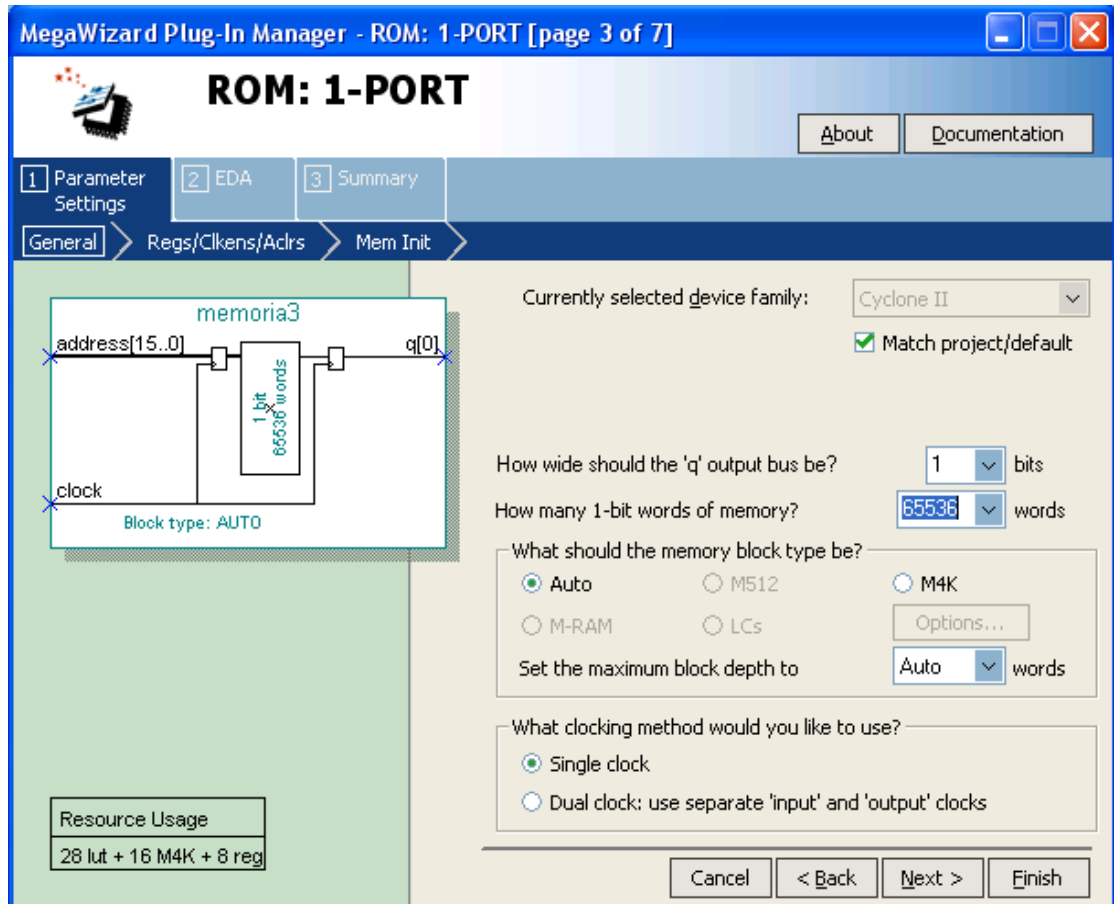


FIGURA4.16. MEGAWIZARD PLUG-IN MANAGER: ROM 1 PORT

En la página siguiente se elimina el Clock para el puerto de salida como se muestra en la figura 4.17. Luego en la siguiente pantalla se especifica el nombre del archivo MIF que se va a utilizar como se muestra en la figura 4.18 para luego en la siguiente pantalla solo presionar NEXT apareciendo la ultima pantalla que muestra en resumen lo que se crea (figura 4.19), presionando

FINISH. Se adjunta el código en VHDL de la ROM que se diseñó al proyecto actual en QUARTUS II.

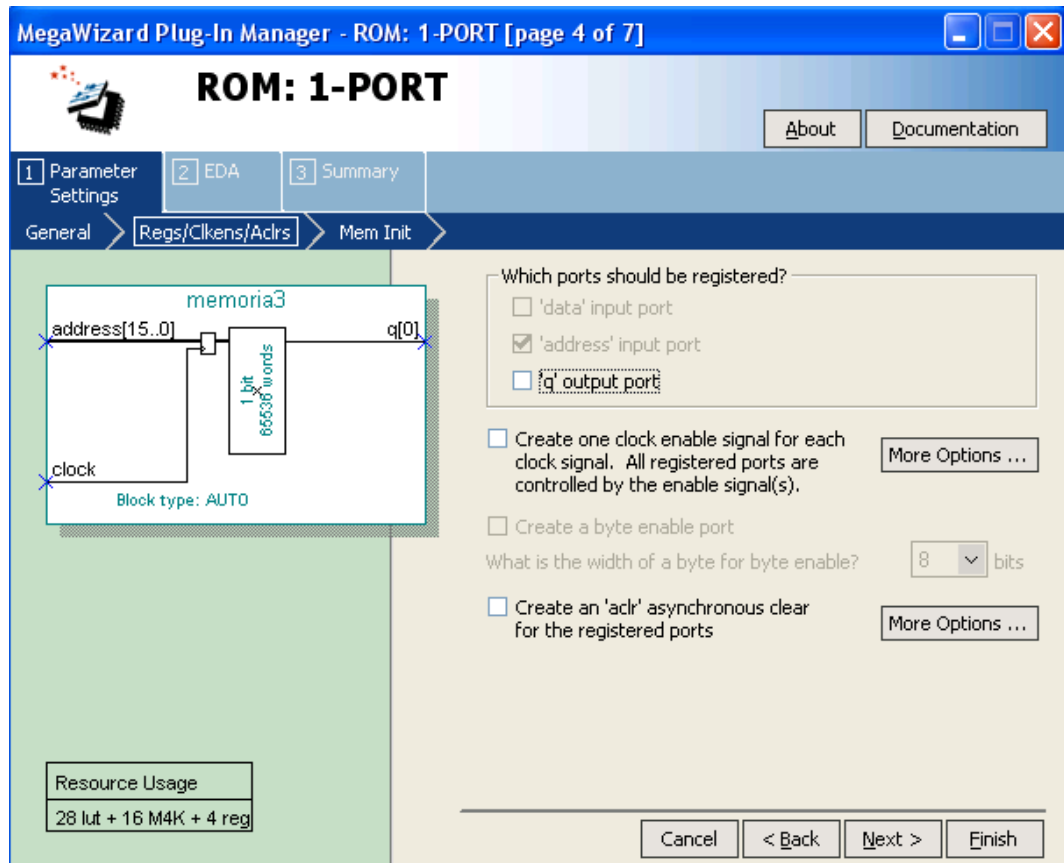


FIGURA4.17. MEGAWIZARD PLUG-IN MANAGER: ROM 1 PORT (PAGE2)

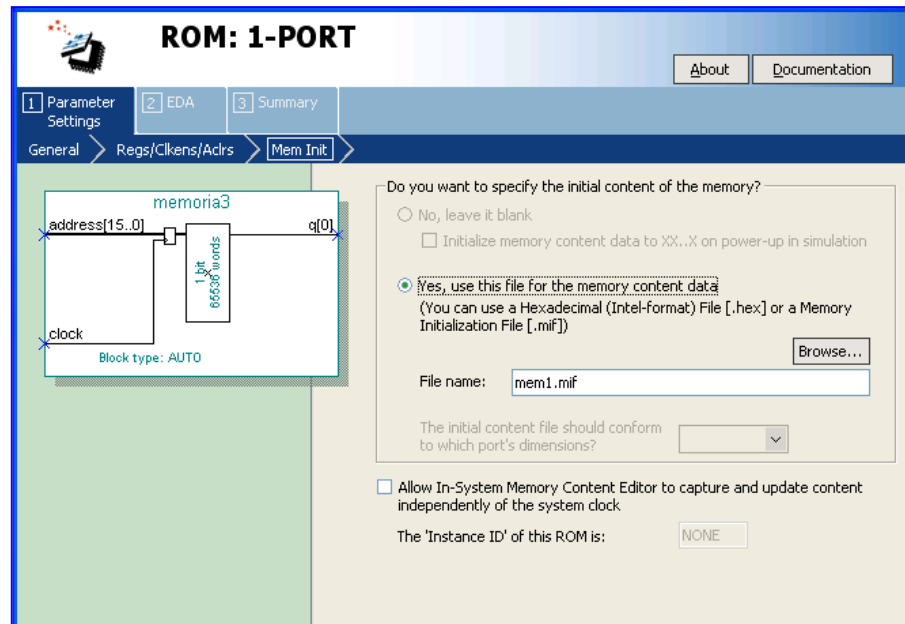


FIGURA4.18. MEGAWIZARD PLUG-IN MANAGER: ROM 1 PORT (PAGE3)

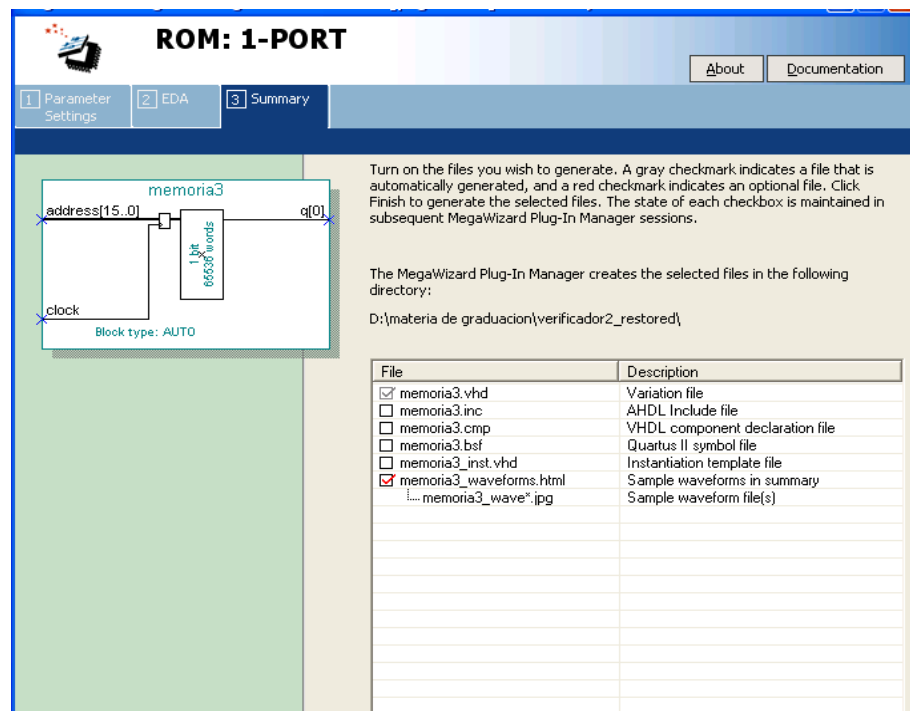


FIGURA4.19. MEGAWIZARD PLUG-IN MANAGER: ROM 1 PORT (SUMMARY)

Se debe tomar en consideración que para cada huella se tiene que crear un archivo MIF y se crearán dos memorias ROM porque solo dos huellas se van a comparar. Un diagrama de bloques general para el proceso sería el siguiente:

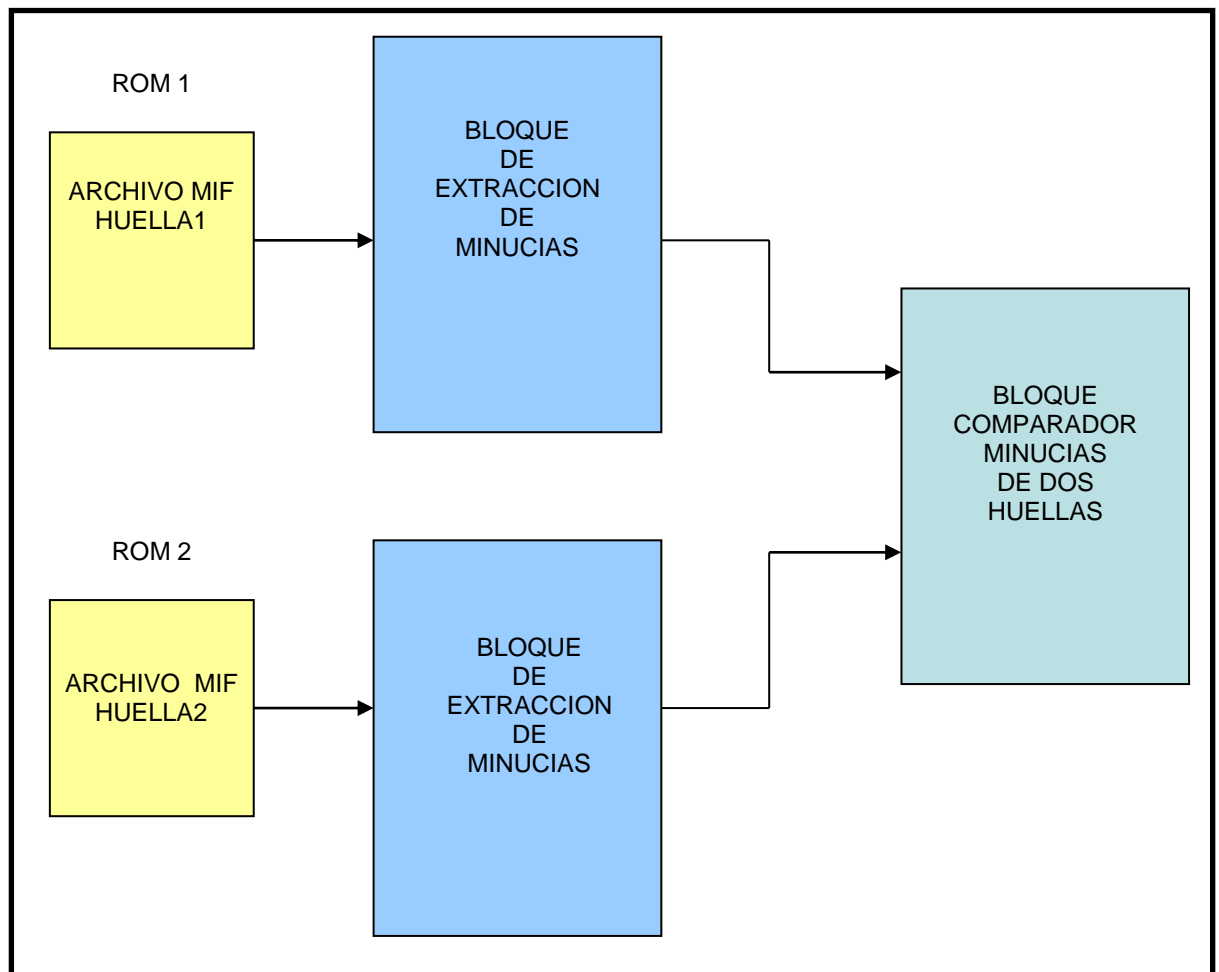


FIGURA4.20. PROCESO DE IMPLEMENTACION EN TARJETA DE2

En donde el bloque de extracción de minucias y el bloque comparador de minucias de dos huellas son algoritmos creados en VHDL.

El bloque de extracción de minucias tiene como objetivo obtener la información de la huella dactilar que se encuentra en la memoria ROM , para analizarla y así determinar el tipo de minucias (por medio del algoritmo de Crossing Number) y coordenadas (de 1 a 215 en X y Y) que existen en la huella. Al término del análisis se tendrá como resultado final cuatro arreglos de coordenadas uno para las coordenadas en X de Terminaciones, otro para las coordenadas en Y, el tercero y cuarto para las coordenadas en X y Y de las Bifurcaciones respectivamente. Los arreglos corresponden a la información que se utilizará para la comparación de las huellas.

A continuación se puede apreciar un diagrama de flujos en donde se esquematiza el bloque de extracción de minucias que para que sea factible se programó el algoritmo mediante máquinas de estados.

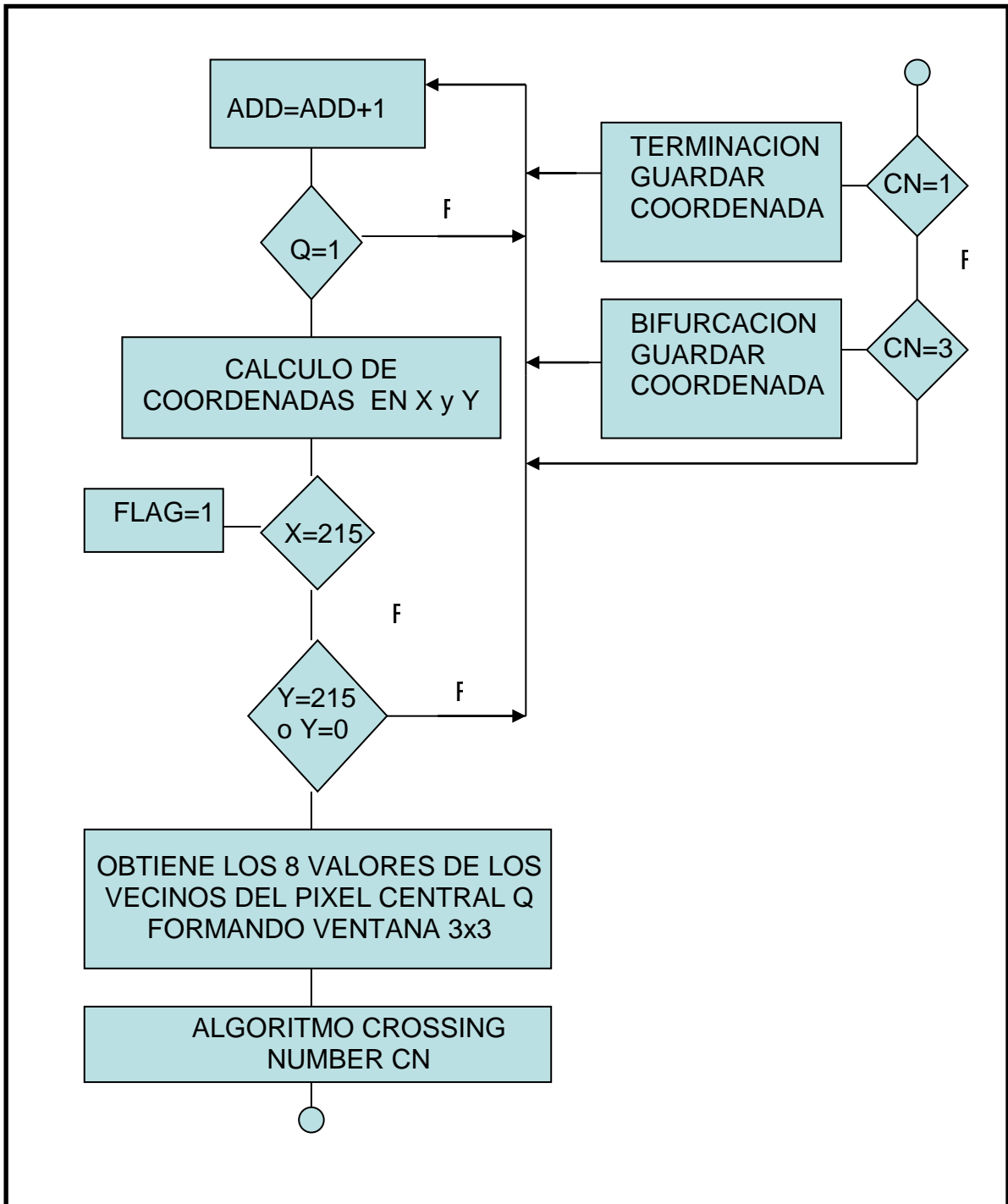


FIGURA4.21. PROCESO DE EXTRACCION DE MINUCIAS

Donde ADD es la dirección de la memoria ROM, Q es el dato que se encuentra en memoria en una dirección específica en donde solo se toman en consideración el valor de 1, FLAG es una señal que indica que se terminó de analizar todos los datos de la memoria ROM, CN es el resultado de aplicar el algoritmo de Crossing Number en donde se indica que tipo de minucia se encuentra presente.

Cabe destacar que los datos de la huella en la memoria ROM se encuentran ordenados desde la dirección 0 hasta 46655 pudiendo calcular las coordenadas desde (0,0) hasta (215,215) al obtener el cociente, de dividir la dirección entre 216, para el caso de la coordenada en X y para la coordenada en Y sería la resta entre la dirección de memoria con la multiplicación entre la coordenada en X y 216, se lo puede apreciar en la ecuación 4.1.

$$\begin{aligned} COORDENADA(X) &= \frac{DIRECCIONMEMORIA}{216} \\ COORDENADA(Y) &= DIRECCIONMEMORIA - (COORDENADA(X) * 216) \end{aligned}$$

ECUACION 4.1. COORDENADAS DEPENDIENTE DE DIRECCION DE MEMORIA

Se debe tomar en consideración la importancia de la exclusión del estudio a los bordes, que en este caso sería cuando la coordenada Y de un píxel

en análisis toma el valor de 0 y 215, porque no se puede aplicar el algoritmo de Crossing Number a un píxel que sea parte del borde ya que al ejecutarlo se evalúa los alrededores formando una ventana de 3x3 con el píxel en análisis en el centro imposibilitando la formación de la ventana con un píxel del borde como central.

El proceso se lo hace para las dos huellas que se van a comparar, por lo que tendremos 4 arreglos para cada huella, siendo información suficiente para elaborar el algoritmo de comparación. En el algoritmo de extracción de minucias existe un FLAG que indica que el proceso de extracción esta terminado, por lo que cuando obtenga el valor de 1 para cada una de las dos huellas, comenzará la etapa de comparación que se muestra en el siguiente diagrama de flujo.

puede verificar en el diagrama de flujo es que se realizan las operaciones del valor absoluto de la resta entre coordenadas de una huella con otra. Se obtuvo el mejor resultado al comparar la cantidad de minucias coincidentes con el 50 por ciento de la cantidad de minucias existentes en las huellas.

Se realizaron pruebas con 9 huellas dactilares de diferentes personas donde se plasmaron 2 de cada uno, teniendo 18 impresiones en total. Se realizó la comparación de huellas abarcando todas las posibilidades y obteniendo los siguientes resultados experimentales:

TOTAL DE HUELLAS	: 18
TOTAL DE COMPARACIONES	: 153
FALSOS RECHAZOS	: 3
FALSAS ACEPTACIONES	: 7
% ERROR	: 6,54 %
EFICIENCIA	: 93,46%

Se pudo determinar experimentalmente la cantidad de terminaciones y bifurcaciones obtenidas por medio de los resultados proporcionados por el algoritmo de extracción de minucias y por medio de la visualización de las

minucias de la imagen de las huellas adelgazadas en matlab. Los resultados se aprecian en las siguientes tablas:

ALGORITMO EXTRACCION DE MINUCIAS		
HUELLAS	TERMINACIONES	BIFURCACIONES
HUELLA1_1	42	32
HUELLA1_2	42	48
HUELLA2_1	22	27
HUELLA2_2	28	25
HUELLA3_1	52	41
HUELLA3_2	44	40
HUELLA4_1	40	25
HUELLA4_2	32	29
HUELLA5_1	55	71
HUELLA5_2	64	75
HUELLA6_1	33	33
HUELLA6_2	35	33
HUELLA7_1	33	58
HUELLA7_2	20	45
HUELLA8_1	36	35
HUELLA8_2	23	23
HUELLA9_1	41	38
HUELLA9_2	35	42

TABLA 4.1. CANTIDAD DE MINUCIAS, RESULTADOS DEL ALGORITMO DE EXTRACCION

VISUALIZACION EN MATLAB		
HUELLAS	TERMINACIONES	BIFURCACIONES
HUELLA1_1	42	32
HUELLA1_2	42	43
HUELLA2_1	22	24
HUELLA2_2	28	19
HUELLA3_1	52	35
HUELLA3_2	44	40
HUELLA4_1	42	23
HUELLA4_2	32	26
HUELLA5_1	55	66
HUELLA5_2	66	66
HUELLA6_1	33	30
HUELLA6_2	35	31
HUELLA7_1	33	54
HUELLA7_2	20	45
HUELLA8_1	36	29
HUELLA8_2	23	23
HUELLA9_1	41	36
HUELLA9_2	35	39

TABLA 4.2. CANTIDAD DE MINUCIAS, RESULTADOS VISUALES

Comparando las dos tablas, se puede obtener el porcentaje de eficiencia de la detección de terminaciones y bifurcaciones para cada huella, la misma que se puede visualizar en la siguiente tabla.

% DE EFICIENCIA EN LA EXTRACCION		
HUELLAS	TERMINACIONES	BIFURCACIONES
HUELLA1_1	100	100
HUELLA1_2	100	89,58
HUELLA2_1	100	88,88
HUELLA2_2	100	76
HUELLA3_1	100	85,36
HUELLA3_2	100	100
HUELLA4_1	95,23	92
HUELLA4_2	100	89,65
HUELLA5_1	100	92,95
HUELLA5_2	96,96	88
HUELLA6_1	100	90,90
HUELLA6_2	100	93,93
HUELLA7_1	100	93,10
HUELLA7_2	100	100
HUELLA8_1	100	92,85
HUELLA8_2	100	100
HUELLA9_1	100	94,73
HUELLA9_2	100	82,85

TABLA 4.3. EFICIENCIA EN LA EXTRACCION DE MINUCIAS

Se determina el promedio en los porcentajes dándonos un resultado de 95,63% que es la eficiencia del algoritmo de detección de minucias.

A continuación se muestra dos impresiones de una misma huella que fueron utilizadas en el banco de pruebas, junto con sus respectivos imágenes binarizadas y adelgazadas, resaltando de color amarillo las terminaciones y de color verde las bifurcaciones.



FIGURA4.23. HUELLA1_1 RESOLUCION 500DPI

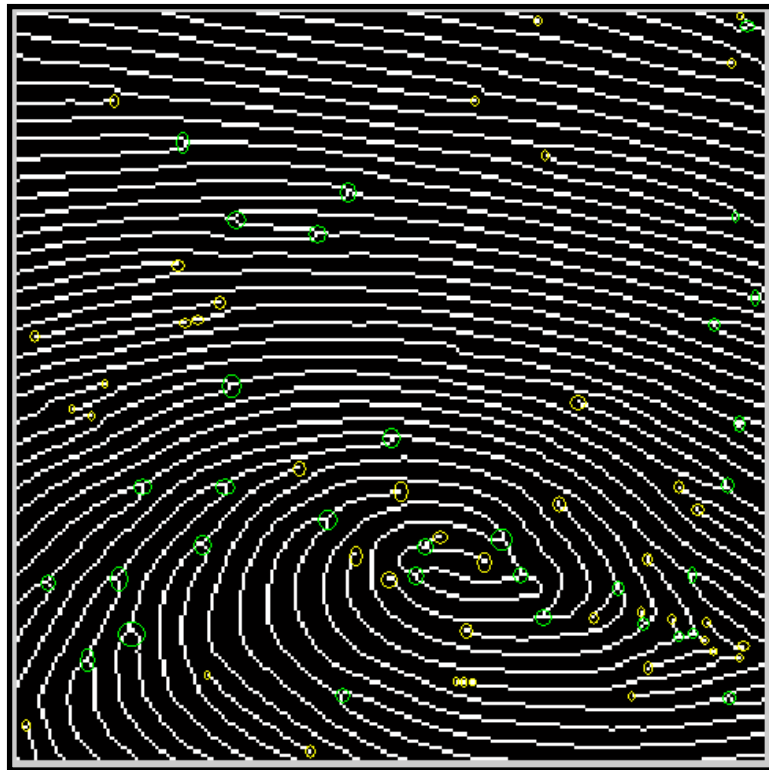


FIGURA4.24. HUELLA1_1 CON MINUCIAS RESALTADAS



FIGURA4.25. HUELLA1_2 RESOLUCION 500DPI

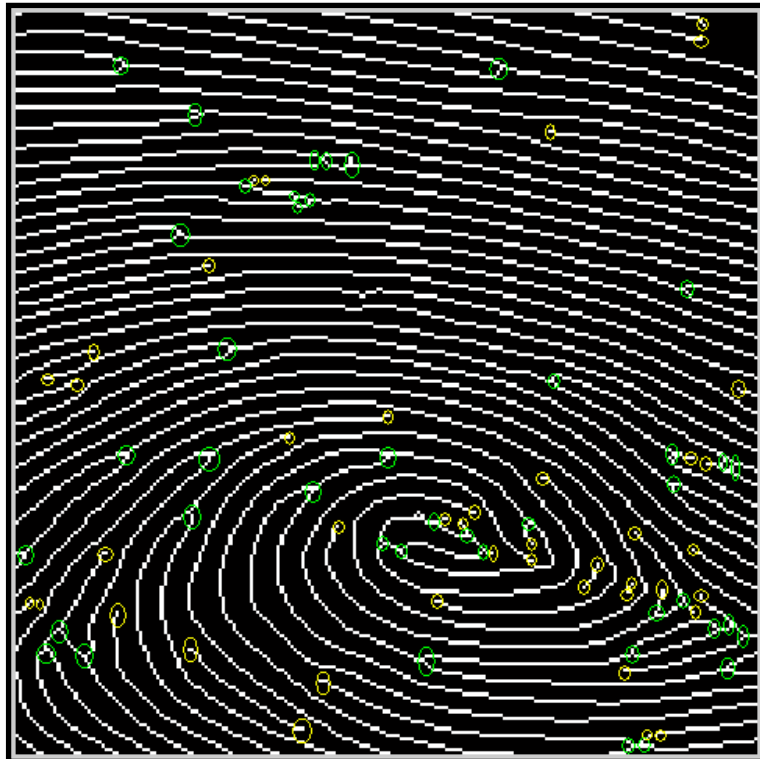


FIGURA4.26. HUELLA1_2 CON MINUCIAS RESALTADAS

El algoritmo de extracción de minucias determina la cantidad de minucias y coordenadas de cada una, que poseen las dos huellas, siendo la información necesaria para la comparación de ambas. Se procedió a comparar las dos huellas en forma manual y por medio del algoritmo creado. La primera se realizó observando las imágenes adelgazadas, determinando la posición de cada minucia en cada huella y luego compararlas. La segunda se realizó visualizando en la tarjeta DE2. Los resultados se observan en la siguiente tabla.

	TERMINACIONES	BIFURCACIONES
HUELLA1_1	42	32
HUELLA1_2	42	48
MINUCIAS IGUALES (OBSERVADAS)	29	26
MINUCIAS IGUALES (ALGORITMO)	32	26
% EFICIENCIA	90,62%	100%

TABLA 4.4. RESULTADOS DE COMPARACION DE DOS HUELLAS

Si promediamos las eficiencias obtendríamos que el algoritmo de comparación de minucias tiene un 95,31% de eficiencia.

CONCLUSIONES Y RECOMENDACIONES

Las conclusiones son:

1. Se pudo observar que las imágenes resultantes del proceso en Matlab no son lo bastante satisfactorias en cuanto a similitud con la imagen escaneada, ya que después del adelgazamiento se producen una cantidad significativa de falsas minucias.
2. Se observó que en algunos casos, las imágenes adelgazadas de dos huellas plasmadas con el mismo dedo difieren en la cantidad y coordenadas de las minucias (lo que en teoría deberían ser idénticas), viéndose reflejado en los 3 falsos rechazos obtenidos en las pruebas.
3. La existencia de una cantidad considerable de falsas minucias hace que incremente la probabilidad de similitudes entre dos huellas distintas, lo

que se vio reflejado en las 7 falsas aceptaciones obtenidas en las pruebas.

4. Los algoritmos de extracción y comparación de minucias poseen un porcentaje de eficiencia mayor al 95% en cuanto a la determinación de coordenadas y cantidad de características en una huella, cantidad de minucias iguales en la comparación de dos huellas, relacionándolas visual y manualmente con la imagen final procesada en matlab.
5. Después de varias pruebas se pudo definir que el valor óptimo para determinar la igualdad entre dos características (proceso de comparación de minucias) es 15, en donde el resultado (en valor absoluto) de restar dos minucias del mismo tipo debe ser menor o igual al valor optimo para que las dos minucias sean consideradas iguales.
6. Se determinó, de acuerdo a pruebas realizadas, que para alcanzar la mayor eficiencia en el proyecto, la cantidad de minucias iguales (de cada tipo) entre dos huellas debe ser mayor o igual al 50% de la cantidad de minucias (de cada tipo) de las dos huellas para que sean consideradas como iguales.

7. Se comprobó que para crear algoritmos complejos lo más adecuado es utilizar una maquina de estados por su flexibilidad en el momento de realizar operaciones con muchos elementos de memoria de distintas direcciones.
8. Se verificó que trabajar con direcciones de memoria optimiza recursos de hardware en el momento de analizar imágenes.

Las recomendaciones son:

1. Se mejoraría la eficiencia si se implementara un algoritmo de eliminación de falsas minucias ya que lo más significativo del error obtenido deriva de las falsas aceptaciones.
2. Se recomendaría utilizar la medida real de la huella escaneada a una resolución de 500 DPI y no reducirla a un tamaño de 256x296 píxeles, como en nuestro caso ya que se estaba limitado por el tamaño máximo de los archivos de memoria de 256x256 píxeles, para el procesamiento en Matlab ya que distorsiona un poco los detalles de la imagen.

3. Se sugiere utilizar un scanner de huella dactilar y no una almohadilla entintada ya que el scanner genera una imagen nítida, libre de ruido, muy diferente a la que se puede obtener mediante el uso de una almohadilla entintada que aunque es de uso exclusivo en dactilografía siempre genera una imagen con manchas.

ANEXOS

Los programas de Matlab para el mejoramiento de huellas se obtuvieron de los siguientes links:

Link principal: <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/GreyTrans/normalise.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/ridgesegment.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/ridgeorient.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/plotridgeorient.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/ridgefreq.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/freqest.m>

<http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/FingerPrints/ridgefilter.m>

Los programas hechos en Matlab y en VHDL serán adjuntados en un CD.

REFERENCIAS BIBLIOGRAFICAS

THAI, Raymond, "Fingerprint image enhancement and minutiae extraction" , (disponible en: <http://www.csse.uwa.edu.au/~pk/studentprojects/raymondthai/RaymondThai.pdf>), 2003.

LOPEZ GARCIA, Juan, "Algoritmo para identificación de personas basado en huellas dactilares", (disponible en: http://upcommons.upc.edu/pfc/bitstream/2099.1/8082/1/proyecto_final_de_carrera.pdf), Consultado en el mes de junio del 2010.

KUONQUI BRAVO, Gira y SAN-WONG MACIAS, Hilda, "Extracción de características y comparación de una huella digital", ESPOL, 2009.

MALTONI, Davide, "Handbook of the fingerprint recognition", (disponible en <http://books.google.com.ec/books?id=1Wpx25D8qOwC&printsec=frontcover&dq=handbook+of+fingerprint+recognition&source=bl&ots=9vTZ4Pjpa6>

&sig=T7wmkfqUs4mYwGCvsfSsin2ZHiA&hl=es&ei=_PYITbPNKYGdlged
qNT8AQ&sa=X&oi=book_result&ct=result&resnum=2&ved=0CDcQ6AEw
AQ#v=onepage&q&f=false), 2009.

MALTONI, Davide, "A tutorial of Fingerprint recognition" , (Disponible en:
http://www.wepapers.com/Papers/15965/A_Tutorial_on_Fingerprint_Recognition), Consultado en el mes de junio del 2010.

HONG, Lin, WAN, Yifei y JAIN, Anil, "Fingerprint image enhancement:
algorithm and performance evaluation",
(Disponible en:
<http://ai.pku.edu.cn/aiwebsite/research.files/collected%20papers%20-%20fingerprint/Fingerprint%20image%20enhancement%20-%20algorithm%20and%20performance%20evaluation.pdf>), 2002.