Hardware/Software Architectures for Iris Biometrics

Autora: Judith Liu Jiménez

Director: Raúl Sánchez Reíllo

Departmento de Tecnología Electrónica

Universidad Carlos III de Madrid

Leganés, Noviembre, 2009

Tesis Doctoral
Hardware/Software Architecture for ID Tokens based on Iris Recognition

Autora: Judith Liu-Jimenez
Director: Raul Sanchez-Reillo

Firma del Tribunal Calificador:

	Nombre y Apellidos	Firma
Presidente		
Vocal		
Vocal		
Vocal		
Secretario		

Calificación:

Leganés,

A Santi,

A mi familia,

A todos los que me habéis levantado para llegar hasta aquí

Acknowledgements

Cuando se termina un proyecto como este y se mira atrás, una se da cuenta de la cantidad de personas que me han estado ayudando a lo largo de estos años a hacerla. Existe un refrán que dice ”de bien nacidos es ser agradecido”, y es en este lugar de la tesis donde he de hacerlo: habéis sido muchos y espero no dejarme a ninguno en el tintero porque sin vosotros esto no hubiera sido posible.

Al primero que tengo que darle las gracias es a mi director de tesis, a Raúl. Durante estos años hemos aprendido mucho los dos ya no solo de biometría, de co-diseño o de todos los temas que hemos tratado aquí, hemos aprendido a respetarnos, a valorarnos. Gracias Raúl por estos años de mutuo aguante.

A Santi, mi compañero, porque me has levantado todas las veces que me he caído y cuando no has podido hacerlo te has quedado a mi lado para que no estuviera sola. Por hacerme reír durante tantos años (y los que nos quedan), porque eres lo mejor que me ha pasado nunca.

A mi pequeño Óscar, que ha pasado horas pegado a mí dándome ánimos, sentado sobre mi pie estando conmigo, pero sin molestar, aunque a veces me quites los bolis. !dichoso perro!

A Fernando y a Ángeles por ayudarme a escribir este rollo, porque me habéis enseñado muchas cosas de la vida, no solo a escribir, sino a enfrentarme a las tareas duras y difíciles. Os habéis convertido en dos de mis grandes gurús, gracias porque os he necesitado desde hace mucho, pero por fin habéis llegado a mi vida.

A mi familia, a mis padres y a mi hermana, porque soy lo que soy gracias a vosotros, porque me habéis inculcado esa cabezonería que tengo para hacer las cosas, porque me habéis inculcado el amor al saber.

A mis compañeros de la universidad, en especial a Cris y Almu porque sois geniales chicas, porque me habéis enseñado el valor de la amistad, porque he podido contar con vosotros incluso cuando sufría en silencio, por los abrazos, por los ratos de risas y por todos los momentos que hemos pasado juntas, que espero que sean muchos muchos más. !Vamos Cris que serás la siguiente! No me puedo olvidar de Vinnie, mi corrector, gracias por ayudarme con esto del inglés, y de Rui ”sí, vale, adios”.

Gracias a mis compañeros del GUTI, a los que están y los que ya se fueron. A Belén, a RAM, a los ”melenudos” porque siempre me habéis tratado como uno más aunque estuviera menos, porque siempre habéis dejado lo que estuvierais haciendo para hacerme comentarios sobre mi tesis, porque siempre he sentido que contaba con vuestro apoyo. Entre ellos especiales gracias a Luis Mengibar por solucionarme mis ”potentes dudillas”. Y por supuesto a Óscar, con el que tengo que solucionar unos cuantos problemas existenciales.

My thanks to the Biometric Group at the Polytechnic University of Poland from whom I have learned a lot while working with all of you, from Biometrics to friendship which linked us. Thank you for teaching me that means are not that important but that people indeed are. Thank you for showing me one of the most interesting places I have ever been at and making me feel that Warsaw was my home. Thank you for giving me one of the best summers of my life.

My special gratitude to John Daugman for showing me how probability can be made easy; for demonstrating that the reisal ways are something to learn, that everybody is interesting through the reisal ways. My thanks also for making me believe in research and showing me that a genius is someone that is able to transmit even the most complicated matter as the simplest thing in the world.

Mariano López y Enrique Cantó son los que me han ayudado muchísimo con algunas de las herramientas que he necesitado en esta tesis, y han conseguido que consiguiera hacer funcionar el EDK y lo que es más !comprenderlo! Muchas gracias a los dos por aquella semana en Vilanova, fue un empujón muy fuerte para este trabajo.

Gracias a Anuska y su panda, por aguantarme las tardes en la tienda cuando me veía incapaz con la tesis. Por enseñarme a manejar la máquina registradora y hacer de mí una veterinaria en potencia.

No me puedo olvidar de todos mis alumnos de estos años, en especial de Patricia, Sandra, David, Luis y a los chavales de telemática. Vosotros inconscientemente me habéis mostrado mi camino, porque me habéis hecho descubrir lo que disfruto enseñándoos, aprendiendo de vosotros.

Y por último y por qué no? A mí misma, porque he conseguido levantarme, porque ha sido un trabajo muy duro y en muchos momentos he pensando en abandonar y si estoy escribiendo esto es porque a pesar de los miles de baches !lo he conseguido!

Abstract

Nowadays, the necessity of identifying users of facilities and services has become quite important not only to determine who accesses a system and/or service, but also to determine which privileges should be provided to each user. For achieving such identification, Biometrics is emerging as a technology that provides a high level of security, as well as being convenient and comfortable for the citizen. Most biometric systems are based on computer solutions, where the identification process is performed by servers or workstations, whose cost and processing time make them not feasible for some situations. However, Microelectronics can provide a suitable solution without the need of complex and expensive computer systems.

Microelectronics is a subfield of Electronics and as the name suggests, is related to the study, development and/or manufacturing of electronic components, i.e. integrated circuits (ICs). We have focused our research in a concrete field of Microelectronics: hardware/software co-design. This technique is widely used for developing specific and high computational cost devices. Its basis relies on using both hardware and software solutions in an effective way, thus, obtaining a device faster than just a software solution, or smaller devices that use dedicated hardware developed for all the processes. The questions on how we can obtain an effective solution for Biometrics will be solved considering all the different aspects of these systems.

In this Thesis, we have made two important contributions: the first one for a verification system based on ID token and secondly, a search engine used for massive recognition systems, both of them related to Iris Biometrics.

The first relevant contribution is a biometric system architecture proposal based on ID tokens in a distributed system. In this contribution, we have specified some considerations to be done in the system and describe the different functionalities of the elements which form it, such as the central servers and/or the terminals. The main functionality of the terminal is

just left to acquiring the initial biometric raw data, which will be transmitted under security cryptographic methods to the token, where all the biometric process will be performed. The ID token architecture is based on hardware/software co-design. The architecture proposed, independent of the modality, divides the biometric process into hardware and software in order to achieve further performance functions, more than in the existing tokens. This partition considers not only the decrease of computational time hardware can provide, but also the reduction of area and power consumption, the increase in security levels and the effects on performance in all the design.

To prove the proposal made, we have implemented an ID token based on Iris Biometrics following our premises. We have developed different modules for an iris algorithm both in hardware and software platforms to obtain results necessary for an effective combination of same. We have also studied different alternatives for solving the partition problem in the hardware/software co-design issue, leading to results which point out tabu search as the fastest algorithm for this purpose. Finally, with all the data obtained, we have been able to obtain different architectures according to different constraints. We have presented architectures where the time is a major requirement, and we have obtained 30% less processing time than in all software solutions. Likewise, another solution has been proposed which provides less area and power consumption. When considering the performance as the most important constraint, two architectures have been presented, one which also tries to minimize the processing time and another which reduces hardware area and power consumption. In regard the security we have also shown two architectures considering time and hardware area as secondary requirements. Finally, we have presented an ultimate architecture where all these factors were considered. These architectures have allowed us to study how hardware improves the security against authentication attacks, how the performance is influenced by the lack of floating point operations in hardware modules, how hardware reduces time with software reducing the hardware area and the power consumption.

The other singular contribution made is the development of a search engine for massive identification schemes, where time is a major constraint as the comparison should be performed over millions of users. We have initially proposed two implementations: following a centralized architecture, where memories are connected to the microprocessor, although the comparison is performed by a dedicated hardware co-processor, and a second approach, where we have connected the memory driver directly in the hardware co-processor. This last architecture has showed us the importance of a correct connection between the elements used when time is a major requirement.

A graphical representation of the different aspects covered in this Thesis is presented in Fig.1, where the relation between the different topics studied can be seen. The main topics, Biometrics and Hardware/Software Co-design have been studied, where several aspects of them have been described, such as the different Biometric modalities, where we have focussed on Iris Biometrics and the security related to these systems. Hardware/Software Co-design has been studied by presenting different design alternatives and by identifying the most suitable configuration for ID Tokens. All the data obtained from this analysis has allowed us to offer two main proposals: The first focuses on the development of a fast search engine device, and the second combines all the factors related to both sciences with regards ID tokens, where different aspects have been combined in its Hardware/Software Design. Both approaches have been implemented to show the feasibility of our proposal. Finally, as a result of the investigation performed and presented in this thesis, further work and conclusions can be presented as a consequence of the work developed.

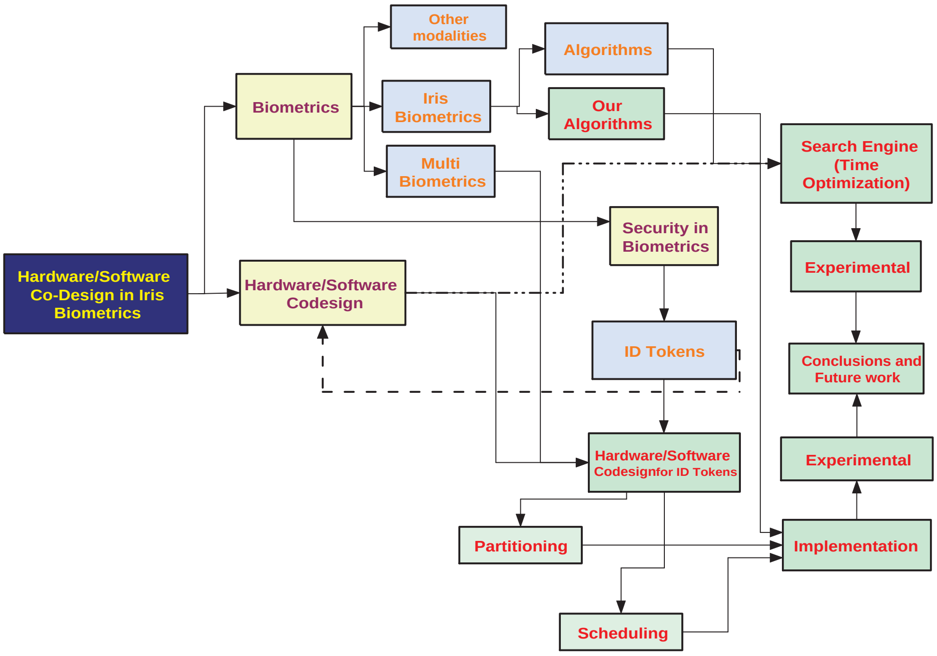


Figure 1: Thesis flow diagram of topics developed

Resumen

Actualmente la identificación usuarios para el acceso a recintos o servicios está cobrando importancia no solo para poder permitir el acceso, sino además para asignar los correspondientes privilegios según el usuario del que se trate. La Biometría es una tecnología emergente que además de realizar estas funciones de identificación, aporta mayores niveles de seguridad que otros métodos empleados, además de resultar más cómodo para el usuario. La mayoría de los sistemas biométricos están basados en ordenadores personales o servidores, sin embargo, la Microelectrónica puede aportar soluciones adecuadas para estos sistemas, con un menor coste y complejidad.

La Microelectrónica es un campo de la Electrónica, que como su nombre sugiere, se basa en el estudio, desarrollo y/o fabricación de componentes electrónicos, también denominados circuitos integrados. Hemos centrado nuestra investigación en un campo específico de la Microelectrónica llamado co-diseño hardware/software. Esta técnica se emplea en el desarrollo de dispositivos específicos que requieren un alto gasto computacional. Se basa en la división de tareas a realizar entre hardware y software, consiguiendo dispositivos más rápidos que aquellos únicamente basados en una de las dos plataformas, y más pequeños que aquellos que se basan únicamente en hardware. Las cuestiones sobre como podemos crear soluciones aplicables a la Biometría son las que intentan ser cubiertas en esta tesis.

En esta tesis, hemos propuesto dos importantes contribuciones: una para aquellos sistemas de verificación que se apoyan en dispositivos de identificación y una segunda que propone el desarrollo de un sistema de búsqueda masiva.

La primera aportación es la metodología para el desarrollo de un sistema distribuido basado en dispositivos de identificación. En nuestra propuesta, el sistema de identificación está formado por un proveedor central de servicios, terminales y dichos dispositivos. Los terminales propuestos únicamente

tienen la función de adquirir la muestra necesaria para la identificación, ya que son los propios dispositivos quienes realizan este proceso. Los dispositivos se apoyan en una arquitectura basada en co-diseño hardware/software, donde los procesos biométricos se realizan en una de las dos plataformas, independientemente de la modalidad biométrica que se trate. El reparto de tareas se realiza de tal manera que el diseñador pueda elegir que parámetros le interesa más enfatizar, y por tanto se puedan obtener distintas arquitecturas según se quiera optimizar el tiempo de procesado, el área o consumo, minimizar los errores de identificación o incluso aumentar la seguridad del sistema por medio de la implementación en hardware de aquellos módulos que sean más susceptibles a ser atacados por intrusos.

Para demostrar esta propuesta, hemos implementado uno de estos dispositivos basándonos en un algoritmo de reconocimiento por iris. Hemos desarrollado todos los módulos de dicho algoritmo tanto en hardware como en software, para posteriormente realizar combinaciones de ellos, en busca de arquitecturas que cumplan ciertos requisitos. Hemos estudiado igualmente distintas alternativas para la solucionar el problema propuesto, basándonos en algoritmos genéticos, enfriamiento simulado y búsqueda tabú.

Con los datos obtenidos del estudio previo y los procedentes de los módulos implementados, hemos obtenido una arquitectura que minimiza el tiempo de ejecución en un 30%, otra que reduce el área y el consumo del dispositivo, dos arquitecturas distintas que evitan la pérdida de precisión y por tanto minimizan los errores en la identificación: una que busca reducir el área al máximo posible y otra que pretende que el tiempo de procesado sea mínimo; dos arquitecturas que buscan aumentar la seguridad, minimizando ya sea el tiempo o el área y por último, una arquitectura donde todos los factores antes nombrados son considerados por igual.

La segunda contribución de la tesis se refiere al desarrollo de un motor de búsqueda para identificación masiva. La premisa seguida en esta propuesta es la de minimizar el tiempo lo más posible para que los usuarios no deban esperar mucho tiempo para ser identificados. Para ello hemos propuesto dos alternativas: una arquitectura clásica donde las memorias están conectadas a un microprocesador central, el cual a su vez se comunica con un co-procesador que realiza las funciones de comparación. Una segunda alternativa, donde las memorias se conectan directamente a dicho co-procesador, evitándose el uso del microprocesador en el proceso de comparación. Ambas propuestas son comparadas y analizadas, mostrando la importancia de una correcta y apropiada conexión de los distintos elementos que forman un sistema.

La Fig. 2 muestra los distintos temas tratados en esta tesis, señalando la relación existente entre ellos. Los principales temas estudiados son la Biometría y el co-diseño hardware/software, describiendo distintos aspectos de ellos, como las diferentes modalidades biométricas, centrándonos en la Biometría por iris o la seguridad relativa a estos sistemas. En el caso del co-diseño hardware/software se presenta un estado de la técnica donde se comentan diversas alternativas para el desarrollo de sistemas empotrados, el trabajo propuesto por otros autores en el ámbito del co-diseño y por último qué características deben cumplir los dispositivos de identificación como sistemas empotrados. Con toda esta información pasamos al desarrollo de las propuestas antes descritas y los desarrollos realizados. Finalmente, conclusiones y trabajo futuro son propuestos a raíz de la investigacin realizada.

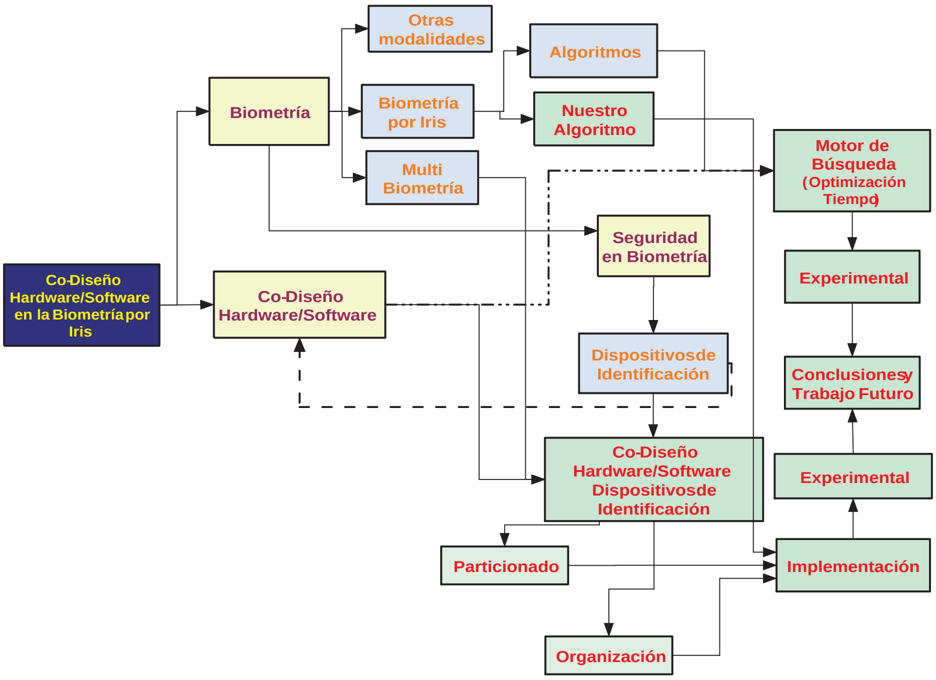

Otras modalidades
Algoritmos
Biometría
Biometría por Iris
Nuestro Algoritmo
Multi Biometría
Motor de Búsqueda (Optimización Tiempo)
Seguridad en Biometría
Co-Diseño Hardware/Software en la Biometría por Iris
Co-Diseño Hardware/Software
Experimental
Dispositivos de Identificación
Conclusiones y Trabajo Futuro
Co-Diseño Hardware/Software Dispositivos de Identificación
Experimental
Particionado
Implementación
Organización

Figure 2: Diagrama de los temas tratados en esta tesis

Contents

List of Figures

Acronyms

AES	Advanced Encryption Standard
AFIS	Automatic Fingerprint Indentification System
ASIC	Application Specific Integrated Circuit
BRAM	Block Random Access Memory Block
CORDIC	COordinate Rotation DIgital Computer
CPU	Central Processor Unit
DET	Detection Error Trade-off
DMA	Direct Memory Access
DSP	Digital Signal Processing
EDK	Embedded Design Kit by Xilinx
EEPROM	Electronic Erasable Programmable Read Only Memory
EER	Equal Error Rate
EMF	Electromagnetic Field
FAR	False Acceptance Rate
FF	Flip-flop
FIFO	First In First Out
FMR	False Match Rate
FNMR	False Non-Match Rate

ACRONYMS

FPGA	Field-Programmable Gate Array
FPU	Floating Point Unit
FRR	False Rejection Rate
FWT	Fast Wavelet Transform
GA	Genetic Algorithm
HDL	Hardware Description Language
IC	Integrated Circuit
ICE	Iris Challenge Evaluation
IP	Intellectual Property
JTAG	Joint Test Action Group
LUT	Look-Up Table
MAC	Multiply Accumulator
MMU	Memory Management Unit
NIST	National Institute of Standards and Technology
OPB	On-chip Peripheral Bus
PDA	Personal Digital Assistant
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PLB	Processor Local Bus
PoS	Point of Service
RAM	Randon Access Memory
RISC	Reduced Instruction Set Computer
ROC	Receiver Operating Curve
ROM	Read Only Memory

RSA	Rivest, Shamir and Adleman encryption protocol
RTOS	Real-Time Operating System
SA	Simulated Annealing
SoC	System on Chip
SoPC	System on Programmable Chip
TS	Tabu Search
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language

Chapter 1

Introduction

Nowadays, the necessity to identify users of facilities and services has become quite important not only to determine who accesses a system and/or service, but also to determine which privileges should be provided to each user. To achieve this identification, Biometrics is emerging as a technology that provides a higher level of security, as well as being convenient and comfortable for the user. Biometrics is currently being applied in many different scenarios, where one of the most common applications is new generation ID documents, such as Citizen ID Cards, or Electronic Passports. In the coming years, Spain will take on an important role in this field, as it will lead the way to the implementation of a new EU biometric passport. In 2010, this passport will be a frontrunner of future biometric control at European member states airports and border checkpoints to restrict illegal immigration: It is one of the reasons that we have centred our attention on Biometrics in this Thesis. Its incorporation into the innovation of border control systems is indispensible to ensure the quality of security.

Most of the research in this field has been focused on the development of algorithms where the performance results have been improved, i.e. the user is recognized with a high level of efficiency and security based on his/her physical traits. All these systems are generally computer based solutions, where the identification process is performed by servers or workstations. However, Microelectronics can provide a suitable solution without the need for complex and expensive computer systems.

Microelectronics is a subfield of Electronics and as the name suggests, is related to the study, development and/or manufacturing of electronic components, i.e. integrated circuits (ICs). With advances in technology, the scale of microelectronic components continues to decrease, achieving small size devices with low processing times, which allow them to be employed in many different fields. We have focused our research within a concrete field of Microelectronics: Hardware/Software co-design. This technique is widely used to develop specific and high computational cost devices. The basis of this technique relies on efficient use of both hardware and software solutions, thus, obtaining a faster device than a software only based solution, or smaller devices that use dedicated hardware developed for all the processes. The questions on how an effective solution can be obtained for Biometrics will be solved by considering all the different aspects of these particular systems. We have carefully studied which processes should be designed and employed using integrated circuits as well as the most effective way to lay out and connect the elements.

Due to the interdisciplinary nature of this Thesis, we have divided this dissertation into several parts, this introduces the reader to the different topics that have been studied, allowing the complete proposal to be clearly understood and well defined. Keeping this in mind, this document has been structured as follows:

- Part I of this document is focused on Biometrics: This part is divided into three chapters, the first dedicated to Biometrics, the second focuses on Iris biometrics and finally, the third describes the security aspects of these systems.
    - In the first chapter, we will introduce Biometrics as a recognition technique, and its use in several application schemes. Some basic biometric terms are also defined for later use.
    - The second chapter is focused on Iris Biometrics. Iris Biometrics has been chosen to verify the proposals made throughout this Thesis. A necessary and deep comprehension of the algorithms used in this modality is necessary to effectively develop any device. In the iris case, several different algorithm proposals have been made, showing the high performance of this technique. The second chapter presents the state-of-the-art of these algorithms along with a detailed description of the algorithms used later on in this Thesis. Finally, a description will be provided on several research projects currently being carried out by researchers to allow interconnection between algorithms and devices.
    - The function of any Biometric system is identification. Due to the nature of these systems they are vulnerable to a wide variety of attacks. Several of these are based on authentication protocols while others are typically performed on the system itself, provoked by the statistical computations these rely on. For these reasons, the third chapter aims to cover the security aspects when dealing with this technology.

- Part II of this document covers one of the key point areas of this Thesis i.e. Hardware/software Co-design.
    - The first chapter focuses on Embedded Systems. At present, these systems are commonly used. The general characteristics and architectures of embedded systems are described in this chapter to provide a clear understanding of the advantages and disadvantages of a Hardware/Software Co-design when compared to other design alternatives.
    - Hardware/Software Co-design will be presented in the following chapter, as well as the work developed in automatic processes.
    - The last chapter of this part focuses on a specific Embedded System used for Biometric purposes, i.e. ID tokens. These tokens are used to store the user's personal information and in some cases, to perform the last step of the Biometric recognition process. We will describe the different alternatives which are commercially available, as well as research work being carried out in this area.
- Part III of this document presents the proposals made in this Thesis.
    - The first proposal is a general Biometric system based on a distributed solution using ID tokens. The functionality of the different elements of the system will be described. The most important contributions have been made regarding the design of the ID token, which has been based on a Hardware/Software Co-design. The techniques developed will be described and verified using an Iris ID token. This development will not only demonstrate the feasibility of our proposal but also its generality, by providing different architectures based on different constraints. This idea will be developed throughout the three chapters of part III: The first chapter indicates design proposals for any ID token based Biometric system, which is independent of the modality chosen. Here a description of the different parts of the system will be provided, as well as a new methodology for token development. The second chapter is focused on the approaches taken in this Thesis regarding Iris Biometrics, and describes the different hardware implementations required to develop an Iris token. Finally, The results obtained are presented in the third chapter, These results have been based on a comparative analysis of the system for different system requirements such as reduced processing time, hardware area or when considering token security.

  - The second proposal is related to the methods required to connect the elements of an Embedded Systems. One of the problems that has arisen in Iris Biometrics is the necessity to develop a device which is capable of computing the maximum number of comparisons in the lowest possible time. We have developed a system for this purpose using different alternatives in the design layout, where the advantages pertaining to a correct connection of the necessary elements is shown.

- Finally, part IV summarizes the conclusions made and described throughout this Thesis, also provided here are indications of future work in this area.

It is the author's intention to further engage in research activities within this field as a result of the investigation performed and presented in this Thesis, where the objective here is to contribute to the EU's goal of improved biometric security at its frontiers for the forthcoming years. Likewise, we are of the firm conviction that any researcher's maximum aspiration is that a Thesis should contribute to executable innovative products and/or services for the benefit of mankind in a highly technological environment.

Part I

Biometrics

Chapter 2

Biometrics

2.1 But what is Biometrics

As children, most of us have played a game called ”who’s who?” In this game, we try to guess our opponent’s identity by means of their physical traits: Are you blue-eyed? Is your nose big? Are you brown-haired? This simple game represents what human beings have been able to do unconsciously from the moment of birth: identify people from their physical traits. From our early days in life, we are able to recognize those who surround us, such as our parents, friends or family; identifying them from their faces, voices or gestures. The necessity for recognition is noticeable during our complete lifespan, to check who are friends among acquaintances or whether we recognize somebody or if he/she is a total stranger; thus, determining how we may interact with others. This function performed naturally by human beings is what Biometrics attempts to do automatically. A biometric system is able to identify a person by his/her face, eyes or even odour.

From the Greek words ”Bios” life and ”metron” measure, Biometrics can be defined as the study of measuring those biological characteristics that make human beings unique; i.e. measurements for recognition purposes. Traditionally, machines perform recognition using knowledge-based techniques (e.g. passwords) and/or token-based techniques (e.g. ID cards) [70] [14]. In cases where high levels of security are required, identification is carried out by combining these two techniques with the intervention of authorized personnel who verify the information provided. However, these techniques have demonstrated several disadvantages that Biometrics is currently trying to overcome. Both conventional solutions mentioned require the user to store and safely keep either knowledge or a token to prove their identity. However, due to the large number of token robberies or guessing of passwords that occur the performance of the identification system is at risk, and also in certain cases where it is not possible to have authorized personnel to perform the identification process. Biometrics, as opposed to the aforementioned, relies on a physical trait of the user, and not on something he/she has, thus, avoiding the already commented risks and the intervention of an agent. With biometric systems, it is not possible to misplace or forget the trait to be measured, as it is an inherent feature of the user [42] [14]. Thus, Biometrics provides a safer way to perform identification.

Table 2.1: User Authentication Systems [109]

		User Authentication		
		Knowledge-based	Object-based	ID-based
Commonly referred to as:		Secret Password	Token	Biometrics
Support authentication by:		Secrecy or Obscurity	Possession	Uniqueness and personalization
Security defense		Closely Kept	Closely held	False proof
Example	Traditional	Combination lock	Metal key	Driver's license
	Digital	Computer password	Key-less car entry	Fingerprint
Security drawback		Less secrecy with each use, easily guessed	Insecure if lost	Difficult to replace

Biometrics allows recognition in the same situations carried out by the previously mentioned systems. Furthermore, Biometrics provides solutions for several situations where conventional systems are unable [131]. Among these situations, the following two are highlighted: the so-called negative recognition and non-repudiation processes. Negative recognition is the process by which a system determines whether a certain individual is indeed enrolled in the system although the individual denies it, e.g. proscription lists or welfare disbursement, where an impostor may attempt to claim multiple benefits under different names. Non-repudiation is a way to guarantee that an individual who accesses a certain facility cannot later deny the use of it, for example when a person accesses a certain computer resource and later claims that an impostor has used it under falsified credentials. In these two situations, and due to security breaches of conventional systems, conventional systems cannot be applied as errors and potential denials make it difficult to prove the authentic identity of the user.

Although Biometrics provides a suitable and reliable solution for some identification scenarios, it is not a panacea for all situations. As a result, other identification problems occur:

- These systems may be subject to malicious attacks, emanating from authentication security breaches or from the possibility of concealing a biometric trait. However, these attacks require much more effort than previous systems, where in comparison, token theft or guessing a password is relatively straightforward.

- These systems do not provide an accurate solution, i.e. they give a probabilistic identification, indicating the grade of similarity between the current trait and the one previously stored. The biometric data obtained is different each time the user accesses the system due to changes in user traits i.e. make-up or aging in face recognition, possible fingerprint scars or voice alterations due to illnesses, etc.

- The cost of these systems is also an important disadvantage. The processing load is much higher and, therefore, requires more powerful machines. Moreover, the human-machine interface, i.e., that part of the system which interacts with the user, often requires high technology devices, thus further increasing the terminal cost.

- As Biometrics relies on a physical trait of the user, private information can also be obtained from the trait under study, this may cause a sensation of intimacy intrusion for the user, to the point that the user rejects this technique. For example, in the case of studying DNA molecules or the retina, these also contain information on inherited malformations or sicknesses, this may provide sufficient reason for some users to reject such modalities.

2.2 The History of Biometrics

Using human features for identification is not a modern technique. First reports on Biometrics are dated as far back as the 14th century, when Chinese merchants used ink to register children's fingerprints for identification purposes [85]. However, this practice was not extended until 1890. In this year, Alphonse Bertillon, a Parisian policeman studied body mechanics and measurements for the first time to help to identify criminals [103]. The police used his method, called The Bertillon Method, until it produced false identifications. The Bertillon Method was then quickly abandoned in favour of fingerprinting, brought back into use by Richard Edward Henry of Scotland Yard. During the 20th century, and due to the evolution of computers, Biometrics has undergone an important evolution. In the early '60s, Biometrics was mainly used for forensic and criminal purposes, with significant progress both in fingerprint and signature identification. Cost reduction in electronic devices and the increase in the computational cost, has lead to the automation of the identification processes, making biometric systems feasible in the current world. At the end of the 20th century, Biometrics became feasible and better known thanks to motion pictures. However, the use of such systems was limited to high security environments, such as military installations or highly sensitive company premises. The massive deployment of these systems has had a decisive starting point after the 9-11 terrorist attacks. Those attempts have proved the necessity for more intensive identification at border controls. From that moment onwards, several biometric systems have been installed at airports to facilitate the identification of travellers for security purposes and for border registration control; for example, at all American airports incoming passengers are requested to provide face and fingerprint samples. Other airports that use biometric systems are Schiphol Airport (Amsterdam), Gatwick and Heathrow Airport (London), which use Iris recognition to identify people at border controls. Several governments have adopted a similar solution for their identification cards, as in the case of the Spanish government [29] which includes biometric information, such as fingerprint or face, on the ID card, also the British ID card provides iris information.

Thanks to the impulse provoked by these developments, the application of Biometrics has increased considerably, it is now frequently found in healthcare environments, at school facilities and used for physical access to several company premises. Another important area where this technique is currently used is in personal computers, where the access is granted to the user after identification using small fingerprint sensors.

2.3 Biometric systems

2.3.1 How a biometric system works?

Biometrics relies on pattern recognition science, which bases its performance on statistics. Pattern recognition attempts to find a new mathematical spaces where different samples from the same object tend to be in the same space area, at a distance from other object areas [120]. In mathematical terms, we can say that we are trying to establish a new space where the intra-class distance (distance between samples from the same object) is significantly lower than the interclass distance (distance between samples from different objects). Fig. 2.1 shows a simple example of the pattern recognition problem. Several samples should be distinguishable, but in the initial space all seem to be mixed. An algorithm is applied to transform all these sets into a new space where the differentiation between each sample is clearer. In this new space, a new sample can be classified into any of the previous groups.

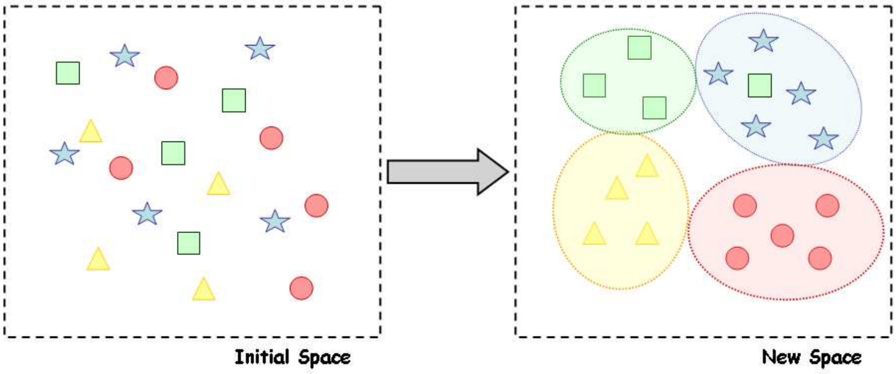


Figure 2.1: Pattern recognition

A biometric system, regardless of the trait being studied, follows the block diagram shown in 2.2 [14] [131].

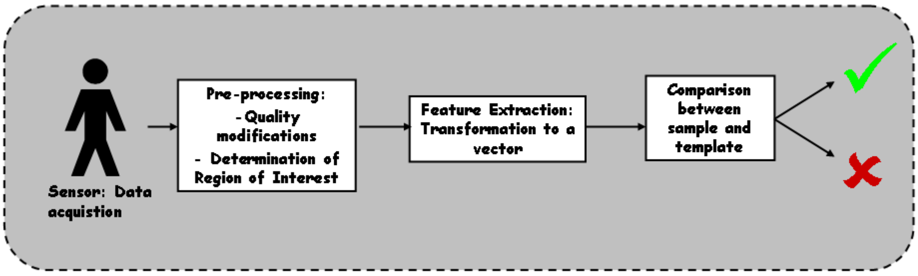

Sensor: Data acquistion
Pre-processing: - Quality modifications - Determination of Region of Interest
Feature Extraction: Transformation to a vector
Comparison between sample and template

Figure 2.2: General biometric schema

The first module in the scheme is the sensor module, biometric reader or scanner. This module is in charge of acquiring the raw biometric trait of the user. This sensor is of great importance as it is the human machine interface and therefore, it is critical for the performance of the complete system. Poor sensor performance may require the trait to be provided several times, causing system rejection by the user . Moreover, such low sensor performance can lead to errors in the results obtained. The sensor technology depends on the trait to be observed and can vary from optical sensors used for fingerprinting, to voice recorders or image cameras. The next module is required for pre-processing of the sample, which is closely related with the next feature extraction block and the initial data quality assessment. This block is responsible for two main tasks:

- To detect whether the quality of the raw sample obtained is sufficient enough to perform further processing. In the case of insufficient quality, another acquisition process is requested. Detection of this quality becomes important when considering its influence on the systems performance; if the algorithm is calibrated for high quality input data, then low quality data may cause errors and vice versa. In this module, processes for increasing data quality, such as equalization, filtering or emphasizing the feature to be detected are also performed.

- The pre-processing itself, within the extraction block, may require previous processing to prepare data from which our representation is to be extracted. Among the possible functions performed by this block, is the segmentation of data to be checked, i.e. detect where a face is in an image, isolate the iris from the rest of an image, detect the core point of a fingerprint, etc.

The block that best represents the recognition problem is the feature extraction block. The function of this block consists of obtaining data which represents univocally the initial data, and that can be used for later comparison. The data resulting from this block is generally called a feature vector, although on some occasions it is only a scalar number. The resulting data also attempts to emphasize the difference between users (in terms of pattern recognition: class). This enhancement facilitates the comparison between the current vector and those that are previously stored in the following block (the comparison module), allowing a decision to be made. As previously mentioned, the result from the comparison is a similarity score. Thus, a threshold can be set to detect if this value is sufficient to ascertain the data as belonging to one user or another. The choice of threshold depends on the systems requirements: if the working environment of the system is moderately restrictive and the number of users to access relatively low, high thresholds are recommended. In this case it is preferable to have several user access attempts rather than a relaxed access control where there is a higher probability of intruder penetration.

Additionally, several authors consider the database as another system block [71]. The database is used to store the users biometric data for later comparisons. The data stored is usually a feature vector or a combination of feature vectors. These stored feature vectors are known as templates. Each template is stored in the system along with other user personal data such as name, nationality, PINs, etc. As will be explained later, not all biometric systems contain a database. In some cases, several databases can be found. Furthermore, users may carry their personal template on a token which is required each time they access the system and where no database is used.

2.3.2 Types of biometric systems

Identification solutions, both conventional and Biometrics or a combination of both, have two possible working scenarios: enrolment and recognition. These scenarios influence the countermeasures taken for security and the physical environment where they can be performed [14] [131].

Enrolment: When a new user wants to be introduced into the system, an enrolment process must be performed. The objective here is to obtain the users biometric template. This template will represent the user and will be used in all of the recognition processes. Enrolment processes can be performed under supervision or in the absence of a human agent, for example when obtaining the Spanish ID cards, a police officer guides you through the steps required to request it, at the same time he/she verifies the data provided. The enrolment in a computer system is opposite to the previous case, as the user usually does this task by him/herself without any help or supervision.

Verification or Identification are the two possible recognition processes which are performed each time a potential user attempts to access a system. Identification is understood as the process of finding the identity of a user where no indications are provided, i.e. it answers the question ”Who am I?” (like in our childhood game). Among N possible users, the system must find the identity of that one person whose data is currently being verified; thus, it is a 1 to N comparison. On the other hand, verification attempts to confirm the identity of a user. The question to be answered by the system is now: “Am I who I claim to be?” Differentiation of these two terms leads to different scenarios, each with their corresponding requirements. For example, when considering forensic Biometrics, identification of a corpse is performed. If the case arises where there is no identification present with the corpse, this process is then performed by comparing data stored in a database. However, if the corpse is found with ID or any other traceable information, there is no need to check all potential identifiers. A relative or close friend can confirm the identity by inspecting the facial characteristics of the corpse.

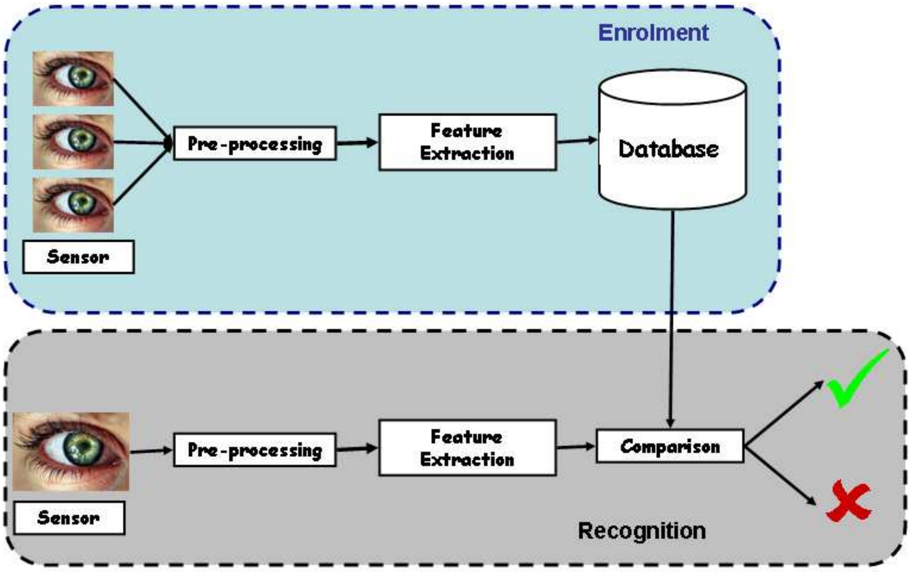


Figure 2.3: Biometric system functions

Identification and Verification functions influence several aspects of the biometric system, such as the algorithm used or the system architecture. The algorithm chosen or trait used determines the interclass distance, i.e. separation among classes. Therefore, for identification, the traits and algorithms with large interclass distances are most suitable, this complicates the comparison process and decreases possible performance errors.

This restriction is not as important for the verification case where the resemblance between the sample and the template does not need to be so high for acceptance, as the comparison is performed 1 to 1, between the current sample and the template of the user whose identity is being claimed. The decision in this case may be less restrictive as no multiple comparisons are carried out. [51].

Regarding the architecture, verification systems allow more diversity than identification systems [140]. In the identification case, the database where the templates are stored should be accessible by the system to perform the comparisons. However, in verification; only the template of the requested identity is required . This necessity marks the architecture the system can support: the continuous access to all templates obliges a centralized architecture for identification, where all the terminals are connected to a single database. This architecture also requires the database to be online permanently, as any system may perform the identification process at any moment. On the other hand, verification only requires access to one template at a time, this permits online and offline solutions, thus allowing a distributed solution. As each template is only used when the user requests access to the system, a central database can be avoided by transferring the templates required to the corresponding user, i.e., each user carries a token where their template is stored, and thus this token must be made available when the verification process is being performed. Therefore, offline solutions can also be used, as no continuous connection to the database is required.

The architecture used influences in many ways the implementation of the system. Centralized solutions provide several advantages which are related to changes and updating of the system, here a modification only requires changes in the central system. This does not affect the user in any way and provides complete transparency. As opposed to these systems, distributed systems present problems when updates or changes are to be performed, where on occasions, this may require changes to be made to each token; thus, the user is required to present the token for this procedure. Most commercial solutions perform updates or changes when the user accesses the system for identification; the system asks the user to wait until the updating is performed [147]. As a result, this solution reduces any interruption to the user.

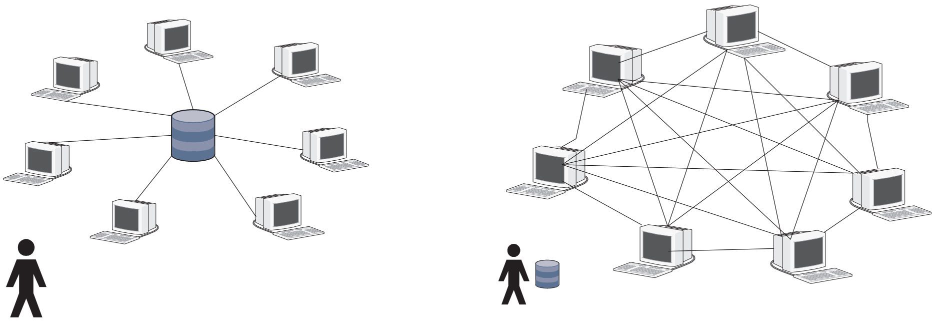


Figure 2.4: Centralized vs decentralized systems

Perhaps the most convenient qualities of token usage are those provided with respect to security and user acceptance. As regards security, both solutions provide weak points. In centralized systems, the database should be highly secured both physically and remotely, as a successful attack can compromise data pertaining to all enrolees. Whereas in distributed systems, acquiring a token containing the information may be much easier. Thus, both systems do not eliminate possible system attacks. However, a successful attack on a centralized system has more disastrous consequences than on a distributed system, this due to the amount of compromised data.

Distributed systems are more accepted by users than centralized systems. Spy movies have shown the possibility of reusing users traits for undesired access. The feeling of privacy invasion due to inherent links between a physical trait and the users private life has provoked several users to reject these systems. This rejection mainly emanates from distrust in the system and data management. Distributed systems provide a solution where the user carries his/her personal data, making him/her feel much more secure, reducing significantly the “big brother” effect.

2.3.3 Performance

As we have mentioned in previous sections, in biometric processes the sample acquired is always different from that previously stored. These samples lead to feature vectors, which in spite of being from a single user, are slightly different. However, if the differences between other users templates are noticeable, the sample being verified can be misidentified with another identity, provoking an error in the system performance. Measuring the errors in Biometrics is important, because these systems are not just subject to errors from pattern recognition problems, but also from the capture process. The detection of these problems can help improve system features and to perform a comparative analysis with other systems. The performance of a biometric system is measured using rates and procedures mentioned in [95] [14]. The most commonly used error rates are:

False Match Rate (FMR): this is the probability that the system incorrectly declares a successful match between the input sample and an incorrect template. It measures the percentage of non-valid matches. If a certain number of attempts are considered, i.e. to determine the result up to a number of attempts which are allowed, this probability is known as False Accept Rate (FAR).

False Non-Match Rate (FNMR): this is the probability that the system incorrectly declares match failure between the input pattern and the matching template from the database. It measures the percentage of valid inputs being rejected. As in the previous case, False Reject Rate (FRR) is the False Non-Match Rate taking into account the number of attempts.

Receiver (or Relative) Operating Characteristic (ROC): In general, the decision taken by the matching algorithm depends on system parameters (e.g. the threshold). In biometric systems, a trade off between the FMR and FNMR can be reached by changing those parameters. The ROC plot is obtained by graphing the values of FMR and FNMR, and by changing the implicit variables. A common variation is the Detection Error Trade-off (DET), which is obtained using normal deviation scales on both axes. This linear graph accentuates the differences for higher performances.

Equal Error Rate (EER) is the rate when both the accept and reject errors are equal. The equal error rate is commonly used when a rapid comparison of two systems is required. The lower the EER value the more accurate the system is considered to be.

Failure to Enrol Rate (FTE or FER): this is the percentage of data input which is considered invalid and therefore, provokes an error when introduced into the system. This failure to enrol occurs when the data obtained by the sensor is considered invalid or of inadequate quality during the enrolment stage, thus denying the user to be enrolled.

Failure to Capture Rate (FTC): This is used for automatic systems, the probability that the system fails to capture a biometric sample when it is presented to the sensor.

Template Capacity: The maximum number of data sets which can be included in the system.

These parameters vary significantly depending on several factors, such as the modality used, the algorithm, and the technology employed. Additionally, some other factors influence them depending on the environment where the system is operational and the system maintenance [132].

Table 1 shows some error rates of different biometric modalities. The data shown in the table has been obtained from several competitions [41], [20].

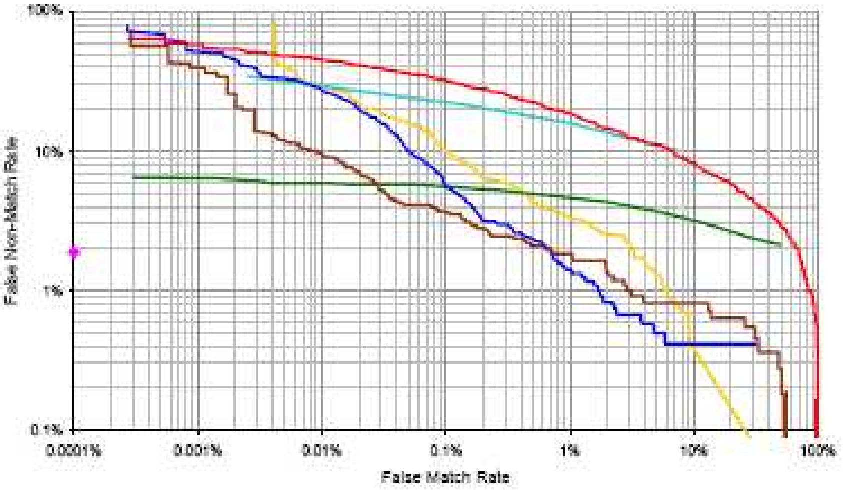


Figure 2.5: DET example [95]

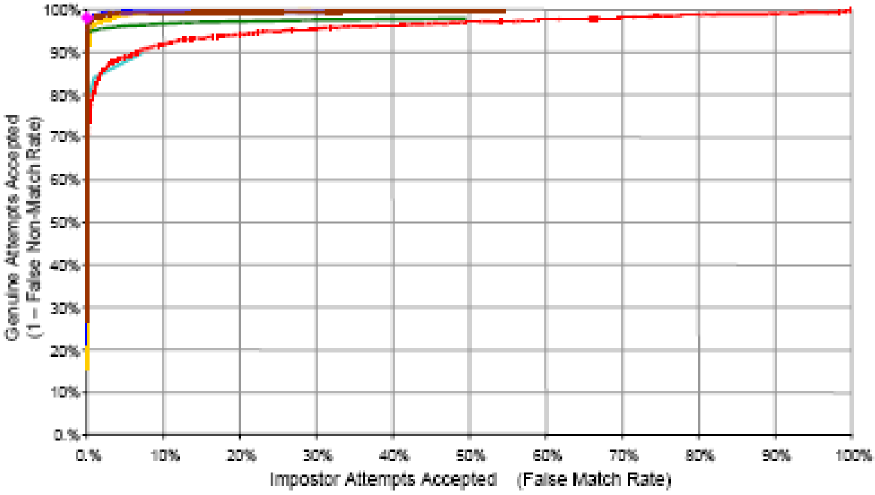


Figure 2.6: ROC example [95]

Table 2.3: Biometric Modalities and their error rates [42]

Modality	EER	FRR	FAR
Fingerprint		2%	0.02%
Iris	2%	0.0001%	0.5%
Face	4%	10%	
Hand	3%	0.3%	0.3%
Voice	7%	7%	
Signature	2.89%	2.89%	

2.4 Biometric modalities

Although previous parameters can help determine the selection of the type of biometric system, the most common parameter considered is the modality studied. The term modality in Biometrics refers to the trait to be recognized by the system, e.g. fingerprint or face. Everyday researchers work on other potential modalities, due to the large amount of these, only a description of the most commonly used and known modalities for recognition will be presented here. Not all human traits can be used for Biometrics. Among all the different traits a human being has, several requirements on the traits have been complied and studied [70] [14], these are:

- Uniqueness, this is directly related to how well the biometric system can differentiate users.
- Universality, this describes how common a biometric trait is among users.
- Permanence, which measures how well a biometric trait can resist aging and other temporal conditions.
- Collectability or ease of acquisition for measurement.
- Performance: in many aspects, such as accuracy, speed and robustness.
- Acceptability: degree of approval for the technology.
- Circumvention: ease of use of a given substitute.

Although most of the characteristics studied do not fulfil all of the above mentioned conditions, they do comply with most of them. Table 2.4 shows qualitatively how different modalities fulfil these features.

Fig. 2.7 shows the impact of the different modalities on the current market. Most commercial solutions are based on fingerprints/AFIS (Automatic Fingerprint Identification System). Although this modality is the most widespread technique, solutions based on face, iris and voice are also found.

The traits studied can be divided into two major groups: physical or behavioural. Physical modalities are those referred to, as the name indicates, a physical characteristic of the user, such as face, fingerprint or hand geometry. Whereas, behavioural modalities are those related to the way a user does something. Examples of behavioural modalities are speech, signature recognition and key dynamics. Behavioural modalities are non-invasive and therefore are more accepted by users. However, these techniques are, in general, easier to falsify and imitation becomes an possible attack.

Table 2.4: Biometric Modalities Comparison [70]

Modality	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	High	Low	Medium	High	Low	High	Low
Fingerprint	Medium	High	High	Medium	High	Medium	High
Hand Geometry	Medium	Medium	Medium	High	Medium	Medium	Medium
Key Dynamics	Low	Low	Low	Medium	Low	Medium	Medium
Hand thermograms	Medium	Medium	Medium	Medium	Medium	Medium	High
Iris	High	High	High	Medium	High	Low	High
Retina	High	High	Medium	Low	High	Low	High
Signature	Low	Low	Low	High	Low	High	Low
Voice	Medium	Low	Low	Medium	Medium	High	Low
Odour	High	High	High	Low	Low	Medium	Low
DNA	High	High	High	Low	High	Low	Low
Gait	Medium	Low	Low	High	Low	High	Medium
Ear	Medium	Medium	High	Medium	Medium	High	Medium

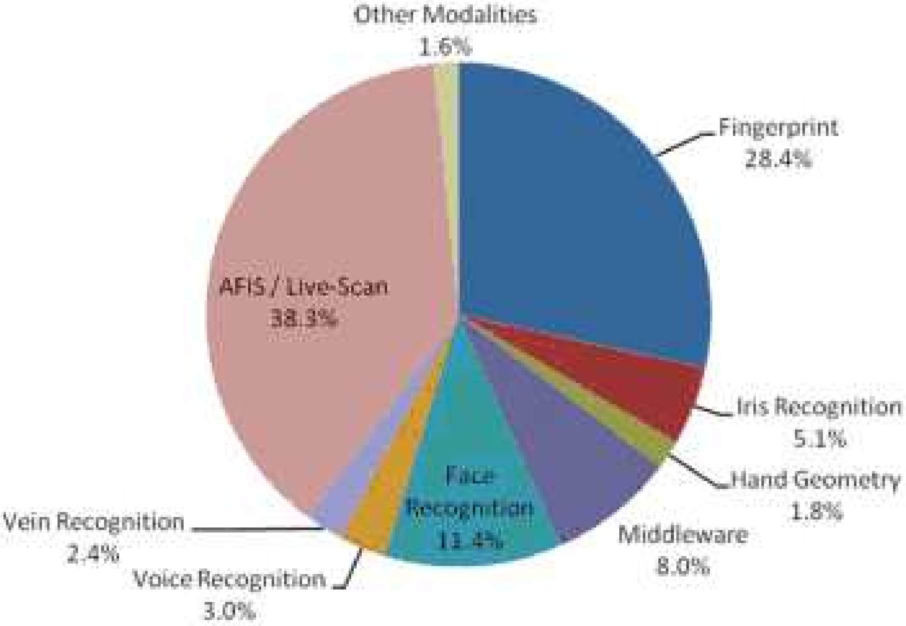


Figure 2.7: Biometric revenues by technology 2009 [57]

2.4.1 Physical modalities

2.4.1.1 Face

Facial recognition is the most intuitive biometric modality, as it is the instinctive manner in which humans identify each other. Face techniques are based on two main ideas: consider the face as a unique organ, and therefore, taking each face as a unique pattern; or measure important points on the face, such as eyes, nose, mouth, etc. and the distance between them [131] [14].

However, this modality has several disadvantages researchers are currently trying to solve, mainly in the pre-processing and quality assessment block. Although the sensor used in this technique does not have to provide high quality images, algorithms are quite sensitive to illumination and surrounding environmental conditions. Potential shadows on the face, movement in the surroundings and the users posture cause serious problems in the recognition or detection of the face. These problems are being solved using 3D images instead of 2D images and by using thermographic images [133] [77] [2]. However, the cost these solutions imply is still relatively high due to the sensor technology and the computational load introduced by the new algorithms.

It is important to remark that a face is not a univocal trait, e.g. twins or people with similar faces can be misidentified. Furthermore artificial techniques can be used to change the structure of the face to make them similar to others, such as make-up or plastic surgery. These problems as well as changes due to aging, lead to errors and decrease the performance of these systems. However, the high user acceptance has made this modality one of the most demanded.

2.4.1.2 Fingerprint

A fingerprint is defined by a pattern of ridges and valleys on the surface of the fingertip [80]. This pattern is developed in the early stage of the foetal development and follows a stochastic process. Therefore, it is considered to be unique for each person and for each finger.

Fingerprint systems can be divided into two different types: those based on macroscopic features and those based on microscopic details. The macroscopic systems base their search on the comparison on region of interest of the fingerprint, meanwhile microscopic solution looks for the distance between ridges or bifurcations.

Fingerprint technology has become the most commonly used modality (almost 80% of the market). This modality has been enforced in several applications thanks to low cost acquisition sensors; fingerprint sensors are now commonly found on laptops and at border controls. The accuracy of these systems is sufficient for authentication and for identification systems as the error rates are low, especially if multiple fingerprints from a single user are provided. However, the main problem associated with these large-scale recognition systems is that high computational resources are required.

Finally, fingerprints are a universal feature. However, a small fraction of the population may have fingerprints unsuitable for recognition because of genetic factors, aging, environmental or for occupational reasons. This problem is usually related to the sensor technology used, as some sensors present problems when acquiring samples from the above mentioned populations.

2.4.1.3 Hand

The hand can lead to three different modalities depending on what is being measured: geometry, palmprint or vascular structure. Each of these modalities relies on the same physical part of the body, however, the trait measured in each case is different; thus, there is a wide range of performance results, and several advantages and disadvantages depending on the detection technique.

Hand geometry

When the measurement considered is based on the shape of the hand i.e. length and width of each finger, this is known as hand geometry [125]. The size of these parts is not univocal and changes slightly with aging or exercise; however, it is still sufficient for performing identification among a reduced population.

The sensors used are mainly low cost devices, such as cameras. The acquisition process consists in taking two photos of the hand, one parallel to the palm and another perpendicular to it which is used to measure width. As a result, these sensors are relatively big, and cannot be embedded in devices such as laptops. Typical applications which use this modality are based on physical access where there is no space limitation. In spite of the low performance and the sensor size required, this modality is widely accepted by users as it is a comfortable method [39].

Palmprint

Considering that the fingerprint pattern is sufficient for recognition, a wider pattern such as the palm of the hand is expected to provide a more accurate recognition system [31] [46]. This is the main idea which has prompted the use of palm recognition, i.e. recognizing people by the valleys and ridges in their palms.

Sensors used for this modality are the same as those used for the fingerprint modality, however, they are bigger; therefore the cost associated with this technique is also greater. High resolution scanners can be used as the sensor for palmprints.

Currently a lot of research work is being carried out on this modality which is combined with hand geometry to increase the performance of the system.

Vascular information

One of the most promising modalities is based on thermographic images of veins located in the hand. Vein development is a stochastic process carried out during gestation, and just as the fingerprint or iris, it is considered unique for each individual. Currently, several sensors can be found on the market such as [113], however, results obtained are far less efficient than expected, this is because the images are highly sensitive to environment variations, especially temperature [136]

2.4.1.4 Eye

In spite of the reduced size of this organ, it provides two reliable modalities: the Retina and Iris.

Retina

The retina is one of the modalities which provides better performance results, however, this technique is not well accepted due to the eye invasion during the acquisition process. The scanners used in this modality must capture the veins located at the back part of the eye, i.e. the retina, for measurement. For this reason, these scanners, besides their cost, acquire data by positioning the user close to the scanner, invading the users intimacy. Also, scanning the retina can provide additional details such as blood pressure illnesses, information which is considered as private by the user. As a result, the information contained within these systems is highly sensitive and lead several users to reject its usage.

Iris

The third most commonly used modality is the Iris. The Iris has demonstrated significant results as regards performance especially in the false acceptance rate case. The development of this modality has increased considerably during the last few years, where now, cost effective commercial solutions are readily available [148] [69]. The research work presented in this Thesis is based on this modality. For this reason, a more extended description of the methods involved can be found throughout this document.

2.4.1.5 DNA

Most biologists claim DNA as a recognition method, however, several obstacles must be overcome for it to be considered for Biometrics, where several of these limitations are unavoidable. Some problems related to this modality are:

- Time required to obtain results. This time is extremely high for recognition purposes as the time depends on chemical reactions. This time can be reduced to days, but, obviously, it isn't sufficient for real time solutions.
- Intimacy intrusion: DNA not only provides information for recognition, but also provides information related to potential illnesses, especially hereditary conditions.
- Similarity between family members: people from the same family share some identical DNA molecules, also, identical twins share the same alleles. In these cases, differentiation between them becomes increasingly difficult.

2.4.1.6 Ear

The shape of the ear is a distinctive trait among human beings. The characteristics of the ear do not change significantly during adult ages. Although this modality is not very well known and its use is not common, certain situations have found that shown this modality is the best method to perform recognition [101]. This is the case for the Spanish Police corps, who are developing a database to perform recognition considering this trait. This process will be used in robbery or raid cases where criminals conceal their faces, although it is still possible to study their ears from CCTV recordings.

2.4.1.7 Odour

In a similar way that dogs identify each other i.e. by smelling, odour systems recognize people by this trait. These systems require electronic noses to capture odours. However, there is a significant increase in the average cost of terminals, where this is due to the acquisition technology used, other odours can significantly affect the systems performance, especially user-related scents such as perfumes or deodorants. For all these reasons, the usage of this type of system is reduced mainly to forensic Biometrics.

2.4.2 Behavioural modalities

Behavioural modalities are based on the data derived from an action performed by the user. The way a user performs an action is mainly based on two aspects: the first, learning based on culture, education or environment issues and the second on physical features of the individual. Modalities based on actions are better accepted than physical ones, as these modalities are always non-intrusive, and therefore, make users feel more comfortable.

2.4.2.1 Voice recognition

Voice recognition is a combination of a physical and a behavioural characteristic. Voice is based on two factors: first, on language and the way of speaking which helps in the mouth disposition for modulation and secondly, in physical traits such as vocal chords or the mouth itself.

Voice recognition has been widely studied for many years. Systems based on this modality can be divided into two main groups: text dependent or text independent [70] [131] [14]. The first group is based on pronouncing a specific sentence or words to the sensor, thus, the user is differentiated from potential intruders. The second group, text independent systems recognize users from their speaking habits when pronouncing words or sentences which change with each access. The algorithms from the first group are more straightforward and easier to implement than the second group. The second group is more secure, as it is not possible to deceive the system using voice recordings.

Voice recognition systems have been widely used, especially in remote applications based on telephonic interfaces. The sensors used are simple and low cost devices, such as microphones. However, their quality, surrounding noise and possible affects due to illness affect the performance results significantly.

2.4.2.2 Signature

For years, signature has been used as a method of identification on paper. Why not for Biometrics? Several aspects of the signature can be examined: measuring the peaks, their inclination or counting the information contained in a determined window.

Signature systems can observe only the result of the action, i.e. the signature. These systems do not require that the signature is made at the time of user identification, thus, it can be used in forensic Biometrics. Unfortunately these systems are subject to signature imitations which, in many cases, can easily be copied. As opposed to this type of system, online systems do not only measure the signature itself but also several other factors which include the inclination of the pen when signing, the pressure exerted, etc. These systems present better security features although they are much more difficult to implement and require special sensors called tablets [131].

In spite of the efforts the users have to make to interact with these systems, they are widely accepted by the population, probably due to social habits. Another interesting characteristic from this modality is that the trait measured can be changed by the user when desired, for example if the signature is captured by an intruder; this is one aspect which is not possible with the rest of the biometric modalities.

2.4.2.3 Gait

Gait refers to the manner in which a person walks. This trait although not very distinctive among users, provides an extra advantage: it can be measured at a distance, avoiding contact with the user. This modality, therefore, is interesting for surveillance applications [70] [131].

These methods typically measure how the human body impacts with the floor considering the way the user distributes his/her weight during each step and the space between each step. Logically, this modality is highly influenced by traumatisms, shoes and the surface being walked upon.

2.4.2.4 Key dynamics

Nowadays, when many contracts or businesses are made over the internet or when using computers, remote recognition is necessary. For this reason, and with no additional sensor other than the keyboard, key dynamics has been proposed. This modality refers to the way a user types on the keyboard, how he/she emphasizes certain keys i.e. the time duration for a certain key is press and the time interval between key strokes [19].

2.5 Multibiometrics

Using one single modality presents several disadvantages [131]:

- The trait studied in some cases is not universal and therefore, becomes impossible to be measured in some individuals.
- The algorithm used does not provide sufficient performance results for identification purposes.
- Indexing in large databases can take a long time according to the modality considered.
- Some biometric modalities are easily forgeable, such as voice or fingerprint recognition.
- Data acquired by the system can be noisy or even corrupt, increasing the failure-to-capture rate. Additionally, sensor malfunction can increase this rate.
- In certain applications, a user is required to be continuously monitored or tracked, in order to verify a successful identification.

To overcome these problems researchers have proposed a new biometric approach known as Multibiometrics. These systems combine several biometric systems using one or more modalities at the same time. For example: in the USA border controls, foreigners are required to provide both face and fingerprint identification, where for the fingerprint modality at least 4 fingerprints are taken and compared.

Several combinations have been made, from using different modalities to using just one modality with different sensors or algorithms. This has lead to the classification of systems into the following groups [130]:

Multi-sensor systems: those which only use one modality, although employ several sensors to capture different samples; for example, a facial recognition system which uses a visible range camera for acquiring a face image which works in parallel with an infrared sensor to acquire subsurface information of the user's face.

Multi-algorithm systems: in some cases, invoking multiple feature extraction and/or matching algorithms on the same biometric data can result in improved matching performance.

Multi-instance systems: These systems acquire data for multiple instances of the same body trait. This can be found in fingerprint and iris systems where the prints from several fingers are required or both irises.

Multi-sample systems: A single sensor is used to obtain multiple samples of the same trait, to check the potential variations in the trait or to obtain a more complex representation of it.

Multimodal systems: These systems rely on detection of the identity by considering several biometric traits, such as combining face recognition with fingerprint. In these systems, it is important to remark that the combined modalities should have similar error rates or the resulting multimodal system will not provide more than satisfactory results. When one modality provides low error rates and is combined with a second modality system with worse rates, it can reduce the performance of the first modality.

The increased system complexity caused by the combination of algorithms, samples or modalities, produces a significant increase in the system computational load: Higher performance devices are required when compared to single modality devices, where these have a greater cost than those used in single modality systems. The data capture equipment in the terminals require, in many cases, several sensors, and therefore, their cost also increases. The computational time necessary to perform identification also increases due to the complexity of the decision.

2.6 Biometrics tendency

The use of Biometrics is increasing everyday. Fig 2.8 shows the evolution for these systems which is expected for the next five years. As it can be seen, the revenue expected will double in value over the next 3 years, turning this science into one of the most promising on the information technology market.

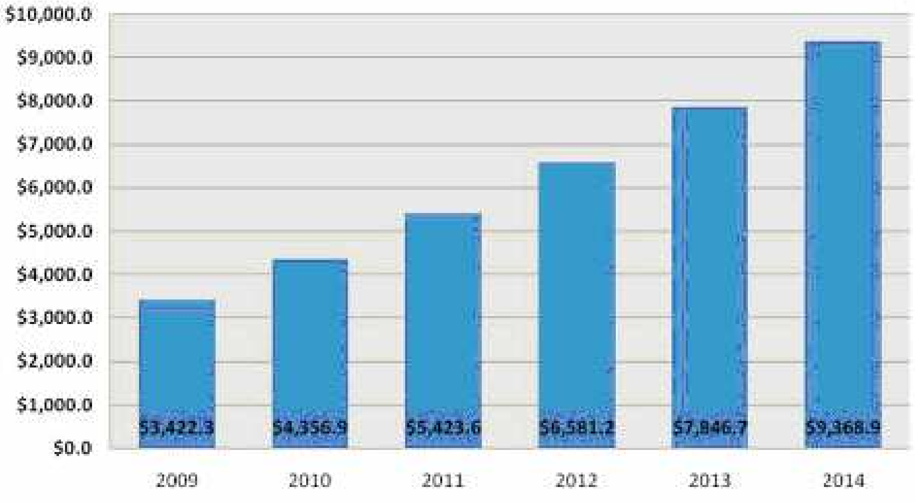


Figure 2.8: Annual biometric industry revenues 2009-2014 [57]

But what is the tendency of Biometrics? What kind of biometric systems do we expect for the future? Several different combinations and implementations of biometric systems are being carried out by researchers and commercially. Most of the work being carried out is on improving system performance, optimizing their algorithms, on the sensors they use and how these sensors interact with the user. This is the case for the previously mentioned face recognition systems, where 2D cameras are being substituted by 3D cameras where it is expected that more data will be acquired which in turn decreases the chances of an identification error.

The system implementation also requires improvements. As we have mentioned in the multibiometrics section, one way of improving the complete system performance is by using different sensors, instances or modalities. However, this setup requires a higher computational load and therefore, high performance devices. The current tendency is to create dedicated systems to perform recognition using hardware with reduced area and for relatively low processing times. Probably one of the most important advances expected is that related to the user-machine interface. Many studies have been performed on the acceptance of such systems by their users, where many cases of users' rejecting these systems exist because of the feeling of invasion of privacy. The user acceptance or rejection to these systems highly influences their implementation and therefore, have a major effect on the expected future deployment. This rejection is clearly influenced by the identity of the verifier and the origin of the user, due to his/her social habits. Biometric systems are widely accepted by users of all nationalities whenever identification is required by governmental agents, such as, at border controls or for fiscal information verification. However, sharing such information between governments is not well accepted which in turn affects the interoperability between systems. The necessity for standardization among biometric techniques and information-sharing in this field is probably one of the most important issues for the future of Biometrics. ISO/IEC JTC1/SC37 [73] are working on providing robust standards in this area. Another important field of research is the security aspect of these systems. The information that is handled is very sensitive as it not only represents univocal and individual information of users but also contains private data.These systems should protect the area where they are applied and also the information they store. For these reasons, several approaches in combining Biometrics with other secure ID techniques have been carried out over the past few years, however, many of these systems are still commercially unavailable.

Such continuous innovation is expected to produce important developments in the commercial market, as demands for reliable secure biometric solutions are required both by public and private sectors. It would not be surprising to have to re-examine proposals for solutions based on combined Biometrics due to the availability of custom-built alternatives offered by commercial product developers on the short-medium timescale.

The brief review presented in this chapter has served to indicate the importance of Biometrics in new control systems required by present day technological global security demands. It has not been intended here to define the most adequate biometric systems to be used, as this is determined by many different factors. However, for the reasons indicated in the following chapter, our work has been motivated by the advances in Iris Biometrics, as it offers excellent security guarantees that will ensure its increasing use in the immediate future.

Chapter 3

Iris Biometrics

3.1 Introduction

Among the many different possible recognition modalities, this Thesis focuses on Iris Biometrics. Although some results of the work presented here may be extrapolated to other modalities, there are several important reasons which have motivated the current research using this method, these are:

- The group this Thesis has been developed in has a very active role in the use of this modality, and have already developed algorithms for such purposes. The more experienced researchers in the group have developed their own algorithms and other previous contributions, encouraging the decision in favour of this modality. Moreover their requests, previous observations and discovered necessities in these systems were the main motivations for this work.
- An iris has features that make this modality appropriate for recognition purposes:
  - Straightforward iris image capture.
  - Its universality, most users have one or two irises; only in rare cases no iris can be provided.
  - The acceptance by users of all ethnics and different cultures.
  - The results obtained by Daugman [25] have shown the reliability of this modality for large scale identification.
- This modality has shown in tests ([108]) the robustness of the algorithms for recognition. At the same time, some of the algorithms involved are relatively straightforward and therefore, do not require high performance devices.

The primary focus of this chapter is on Iris Biometrics, where we initially present the human organ where the iris is located and its main functions. We will then examine the characteristics of the iris that make it suitable for biometric purposes and also delve into the difficulties researchers have been trying to overcome to make its use more comfortable and improve system performance. This is followed by the state-of-art, where the work from other research groups is presented along with the most recent industrial advances.

Throughout the work presented in this Thesis, the algorithm proposed by Sanchez-Avila and Sanchez-Reillo [135] will be described in detail, this was then chosen to validate the proposals which are presented in chapter 8. This chapter concludes with a description of work carried out by different authors to interoperate between different iris algorithms and systems .

3.2 Anatomy of the eye

The human eye is basically composed of three different layers [143]:

- The fibrous tunic or tunica fibrosa oculi is the outer layer and is formed by the cornea and the sclera. The sclera is the layer that gives the eye the white colour. The cornea is a transparent tissue which allows light to enter the eye and protect the inner layers.

- Vascular tunic or tunica vasculosa oculi is the middle layer which includes the iris, the ciliary body and the choroid.

- Nervous tunic or tunica nervosa oculi is the inner layer which includes the retina, composed of photosensitive rods and cone cells which are associated with vision functions.

The iris is situated in the middle layer of the eye between the lens and the cornea and aqueous humor. It consists of a pigmented fibrovascular tissue known as stroma. The stroma connects a sphincter muscle, which is in charge of contracting the pupil, and several dilator muscles, which open it. All these elements form a unique structure of valleys and ridges for each person and are studied in Biometrics for identification.

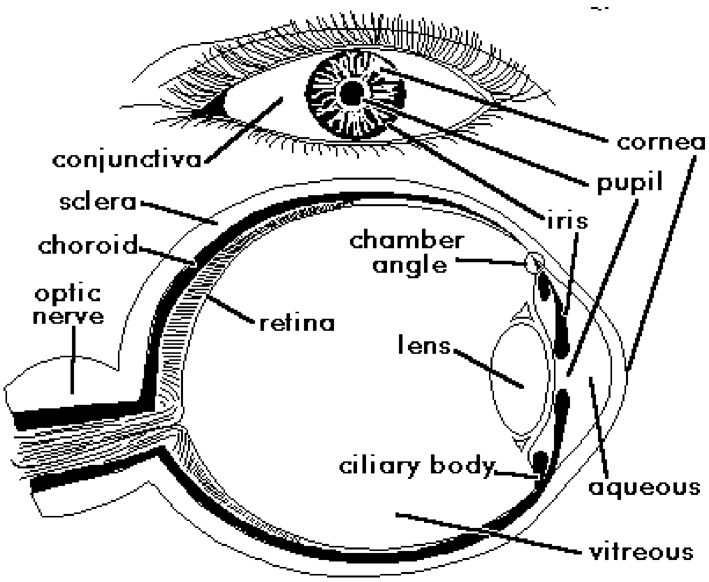


Figure 3.1: Eye anatomy

3.3 Iris for recognition

The iris, due to its situation and anatomy, provides several important characteristics which make it suitable for biometric purposes [28]:

- Its structure is unique, even the two irises from the same person are different. This is due to the iris development during the pre-natal morphogenesis (7th month of gestation), which leads to a random structure which is not genetically dependant.

- Iris patterns possess a high degree of randomness, which make them suitable for large scale identification applications [23]:

  - variability: 244 degrees-of-freedom
  - entropy: 3.2 bits per square-millimeter
  - uniqueness: set by combinatorial complexity

- Due to the location of the iris between the lens and the cornea and aqueous humour, it is protected naturally from possible modifications or accidents. This protection makes it even more difficult to change the iris structure without risking vision damage, and therefore, reducing the possibility of changing it [144], this avoids potential intruders from modifying iris characteristics to defraud the recognition system.

- Although the iris is an internal organ, it is visible thanks to the transparent lens which covers it. Iris patterns can be observed in images which are taken for distances of up to 1 meter between the user and sensor. Therefore, this modality does not need to be intrusive, as no direct contact is required between the user and sensor. Nowadays, sensors exist which can acquire iris images at larger distances [98].

- The iris is present in almost all users; just a small portion of the population do not possess an iris, this is mainly because of aphakia illness [4] or iris removal during cataract surgery.

- The iris structure does not change during the user's lifespan, although its colour does change [23], so lifetime recognition is possible for each user.

- The main function of the iris is to control the light which enters the eye through the pupil. The natural dilations and contractions it makes can be used to prove the natural physiology of it, and thus, liveness detection of the sample examined.

- The computational time required to perform the identification is relatively low, this makes real-time iris detection applications possible.

- Due to cultural issues, iris detection is suitable for use in places where other parts of the body, such as fingerprint or faces are not shown.

- The eye tissue, especially the iris, degrades rapidly after death. This characteristic provides an extra countermeasure against the use of eyes from a corpse to access systems.

Although the iris provides some inherent and natural characteristics which make is suitable for recognition purposes, some problems must be overcome the algorithms used have to deal with them:

- The size of the iris is relatively small. Although images can be acquired from a distance of 1 m, high resolution cameras are required to obtain good performance results at this distance.

- The natural movement of the iris which can be used for liveness detection should be distinguished from natural eye movements, and also separated from face movements, All of these movements increase the difficulty of obtaining a focused iris image.

- The pupil changes with light as well, the iris variation is not elastic. This fact should be considered in measurements taken using the algorithm or removed using the algorithm.

- In most cases, the iris is partially occluded by the eyelid, eye lashes, lenses or reflections. These obstacles should be avoided and considered in the pre-processing algorithms to reduce their influence on the general system performance.

- Some orwellian connotations have been observed in some users, due to the natural confusion between the retina and iris recognition. These connotations may lead to user refusal for such recognition systems.

The use of the iris for recognition has been employed for many years. In 1936, an ophthalmologist, Frank Burst, proposed the idea of using iris structures for recognizing users. However, it was not until 1987 when two ophthalmologists, Leonard Flom and Aran Safir, patented the idea [78] and turned to J. Daugman from Harvard University to implement the first algorithm. J. G. Daugman presented the first algorithm in 1993 which allowed user recognition from iris characteristics. The results he presented were promising, and identified the Iris as one of the best performance biometric modalities (Fig. 3.2).

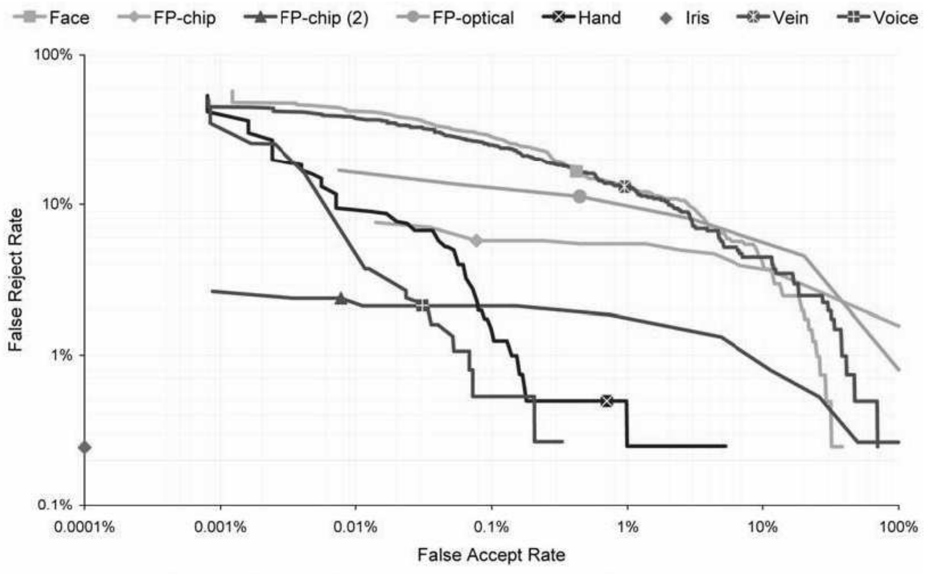


Figure 3.2: Comparison of Iris Biometrics with other modalities [95]

After this first algorithm and due to its performance results, several proposals have been presented for this modality. In the following section, we will discuss the most relevant of these, this is followed by a description of the algorithm used in this Thesis. It is important to emphasize that although the Thesis work is based on Iris Biometrics, the algorithms used have not been developed as part of this work but have been provided by the authors' research group and based on the work of J.Daugman. However, an in depth knowledge of the algorithms is required to effectively implement them for the work presented in this Thesis.

3.4 State-of-art

From a conceptual point of view, an iris recognition system has the same block diagram as any other biometric modality (Fig. 3.3).

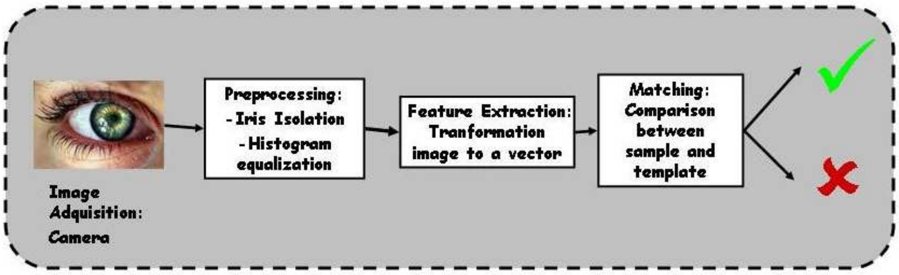


Image Acquisition: Camera
Preprocessing: - Iris Isolation - Histogram equalization
Feature Extraction: Tranformation image to a vector
Matching: Comparison between sample and template

Figure 3.3: General scheme of an iris biometric system

First, an image of the iris is captured by the sensor, this process is done using cameras specifically set up for this purpose (Section 3.4.1). After capturing the eye image, a **pre-processing** block is used to perform the processes required for the subsequent blocks. Among these processes are:

- Locating the iris in the image by segmentation, detecting both iris boundaries (iris-sclera and iris-pupil).
- Those required to improve image quality, such as image equalization.
- Other processes which reduce the computational load on the system or make it more straightforward to follow block tasks, such as image resizing or transformations.

Feature extraction consists of transforming the iris into a number of features, denoted as the feature vector or iris code, this represents the iris under study. This transformation is the most characteristic part of the algorithm as it strongly determines the performance. Different algorithms are presented, but all of them attempt to represent the iris structures ridges and valleys using a measurable vector. The feature extraction is not used to detect the colour or size of the iris as these two features change during the user's lifespan.

The code obtained from the feature extraction should represent the iris and be sufficiently different for each user. This differentiation from the sample to any previous data stored, i.e. template, is performed using the **comparison** algorithm. This calculates the distance between the sample and the template and depending on these results determines if the sample belongs to the user whose template is being used for the comparison.

In the following subsections the state of the art for this technology has been developed over the last years. First a detailed description is presented on several commercial systems for acquisition currently available on the market and some of the work that has been carried out on these sensors to increase the user friendliness of this technology. Most researchers have proved their proposals using databases, instead of developing their own sensors. The public database will be presented in the following subsection and shows the differences between different samples and their influence on the algorithm.

This is followed by a review of the biometric algorithms which have already been proposed by different authors, with a description of the different approaches used for data pre-processing, extracting the feature vector and finally the comparison among templates and samples.

3.4.1 Acquisition sensors

As Fig. 3.3 shows the first part of the biometric system is the sensor used to acquire the user raw data, i.e. the human-machine interface. This interface depends on the modality and has a large influence on the systems overall performance [132], [136].

As opposed to general belief, to acquire data in Iris Biometrics, there is no need for a laser-scan of the eye. To obtain an image of the Iris, cameras are used; initial applications worked in the visible range, acquiring colour images of the user's iris. These cameras presented a major disadvantage which had to be solved: lighting. The ambient light of the room where the camera is located may be relatively low thus a flash is required and it is reflected in the user's iris. These reflections influence iris measurements, the part of the iris where this reflection is located is discarded, also the system is not comfortable for users due to the flash. For these reasons, most iris cameras working in the infrared range are now used. The infrared region provides multiple advantages when compared to the visible range: iris ridges are more defined; the border between the iris and the pupil is more pronounced; and the user is not subjected to exposure of annoying flashes.

Other considerations should be made for adequate sensor design: framing iris images without unduly constraining the user, i.e. the user should be able to position himself/herself without the need of an eye piece, chin rest or any other contact positioning device. For this reason, most commercial cameras provide LEDs, LCDs or recorded messages to advise the user of any adjustments or directional movements or by using a mirror where the user is able to see his/her eye reflected in the correct image acquiring position. Table 3.1 shows some of the commercial iris cameras currently available on the market [58].

A lot of research work is being performed on calibration problems associated with these sensors, i.e. the size of the iris is small and therefore, cameras should be calibrated for small objects at a relatively large distance (considering the iris size and in order to avoid user intimacy invasion). Some authors [63] [114] [56] have been working on developing auto-focus systems for these sensors, this removes the requirement of the user to be positioned at a specific point where the sensor is focused-image calibrated. Further work in this area has been presented by an American group [98] where they have developed a sensor which is capable of capturing the iris image while the user is moving. The application of this sensor is straightforward and will aid in reducing long queues in massive identification scenarios such as border controls. However, continuing research work must still be carried out on this topic to achieve high quality images for recognition.

Research work on sensors is becoming a popular endeavour among researchers and the industrial community. Sensors are probably the weakest point of iris biometric systems for two reasons:

- Their influence on the algorithm performance. Initial data provided for the system influences significantly its performance. The image resolution and focus may provoke errors in identification due to the lack of image quality and the absence of clear ridges to measure.

- Since the system is based on a human-machine interface, it should be as comfortable as possible for users. Iris sensors are considered among most users as

Table 3.1: Iris Sensors [23]

Camera	MFG	Features
IrisAccess 3000	LG	Widely deployed Auto-focus Auto-zoom Voice interface Anti-spoofing countermeasures
ET-300	Panasonic	Dual-eye camera voice interface Oblique illumination so eyeglasses need not be removed
ET-330	Panasonic	Fast 1.0 second recognition speed High security, with false acceptance ratio as 1 in 1.2 million Built in support for PROX cards and smartcards Voice guidance recognition procedure Tamper detection
Authenthicam	Panasonic	small, low-cost camera for PC login or database access control
IrisPass-M	Oki	Active-tilt eye-finding camera Automatically adjusting for height and position Voice interface Anti-spoofing countermeasures
Portable Iris Enrollment and Recognition)	Securimetrics	Handheld camera military and police deployments
H100 camera	IrisGuard	Auto-focus Auto-zoom LCD USB-II interface Flexible deployment

intrusive and difficult to use, requiring large amounts of time for users to position themselves in the correct position for identification. These reasons do not encourage many users to use this modality.

For these reasons, much work is being carried out in this field by introducing new approaches which in less time improve the features of the current solutions and therefore it is expected to increase the deployment of this modality [90].

3.4.1.1 Iris image databases

Most of the previously described commercial systems do not provide the images they capture (only a few provide images, thanks to software development kits). For this reason, most researchers have to validate their algorithms using iris image databases. Several databases of this kind are available, however, it must be emphasized how remarkable initial characteristics in image studies determine the algorithms behaviour [132], thus results obtained from one database when compared to another can vary significantly. Even though there is not an international consensus on the databases used to test algorithms, most of the users try to use similar databases in their trials.

The most commonly used iris database is the CASIA database [50] which is made up of three different sets of images of infrared iris images provided by the Center of Biometrics and Security Research in China. As a result most of the images belong to Asian users:

- The first set contains 108 users (both eyes) and 7 samples of each eye, with uniform lighting providing homogeneous grey level images. This database offers the relevant characteristic for the pre-processing that all images undergo. This process has eliminated all specular reflections and provides all pupils with the same colour.

- The second set is formed from approximately 140 users, for 20 samples from each user. In this database the characteristic pre-processing of the previous version has been eliminated. The image area is wider and is not only limited to the eye but also provides information as regards eye lashes, eye lids, etc.

- The third set provides images from 700 users, taken during different sessions and with different cameras. The surrounding environment where the images have been taken also varies from one camera to another, i.e. indoor and outdoor.

Although these databases are the most well-known, there are others available, such as the UBIRIS [35] and the UPOL database [34] where these provide images taken with a standard camera.

Finally, the ICE 2005 database is the only database developed by a government institution [108]. This database was used in 2005 for the first Iris Challenge Evaluation to test several algorithms. This database is formed from 1425 images from 140 right eyes and 1528 images of the left eye from 120 users. The images were taken under different environmental conditions and situations, the database contains images from users with glasses, make-up or even coloured contact lenses all of which make the identification process more difficult.

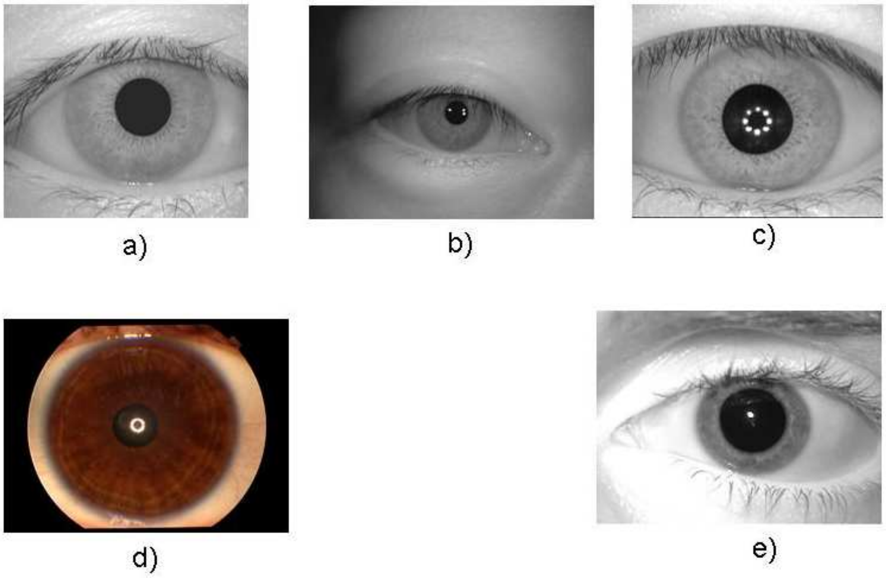


Figure 3.4: Examples of images available from online databases: a)CASIA database v1, b)CASIA database v2, c)CASIA database v3, d)UBIRIS database, e)ICE 2005 Database

Recently, authors have developed a database formed from synthetic iris images which are suitable for use when testing algorithms [158].

3.4.2 Pre-processing block

Segmenting the iris from the rest of the image is the most important part of this block. Segmenting the iris means locating it in the image and discerning which parts of the image belong or not to the iris, eyelids, eyelashes, the pupil and the sclera. This process is performed in several ways, depending on the initial considerations.

The first intuitive approach is to consider the iris as an annulus . However, a later study has shown that the iris cannot be approximated by two concentric circles, as the pupil and iris are not perfectly round. Various different illnesses can provoke different pupil or iris forms. Moreover, if the user does not "stop and stare" at the camera, i.e. looks in a different direction away from the lens, then the shape of the iris and the pupil are elliptical rather than circular. Initial proposals have been made which consider the iris as being surrounded by two non-concentric circles, currently several authors are studying new methods to reconstruct irises which do not fulfil this approach by detecting the gaze angle and distance.

Daugman in his first approach [28] proposed an integro-differential operator, which is based on the difference in the gray levels of the pixels between circles drawn in the image and considers a gaussian smoothing approach.

$$max_{n\triangle r,x_0,y_0}\left|\frac{1}{\triangle r}\sum_k\left\{G_\sigma((n-k)\triangle r)-G_\sigma((n-k-1)\triangle r))\sum I[k\triangle r\cos(m\triangle\Theta)+x_0,k\triangle r\sin(m\triangle\Theta)+y_0]\right\}\right| \quad (3.1)$$

$$max_{n\triangle r,x_0,y_0}\left|\sum_k\left\{\frac{G_\sigma((n-k)\triangle r)-G_\sigma((n-k-1)\triangle r))\sum_m I[k\triangle r\cos(m\triangle\Theta)+x_0,k\triangle r\sin(m\triangle\Theta)+y_0]}{\triangle r\sum_m I[((k-2)\triangle r\cos(m\triangle\Theta)+x_0,(k-2)k\triangle r\sin(m\triangle\Theta)+y_0)]}\right\}\right| \quad (3.2)$$

In the case of the iris outer boundary, the angle from 3.4.2 only varies from -45 to 45 and 135 to 225, in order to avoid eyelids and eyelashes, which usually occlude the upper and lower iris boundaries. After finding the outer boundary, the inner boundary is found within the inside area of the iris, reducing the computational load on the system. Sanchez-Avila *et al.* [135] have used a similar approach, but instead of investigating the difference between circumferences, they search for the maximum difference between lines drawn that cross the complete image. The difference is computed considering intervals of five pixels with an interval of the same size. As with Daugman‘s work, the authors start by searching for the outer boundary within the complete image this is followed by a search for the pupil inside the area delimited by the pupil values.

Several other authors [155],[84],[91],[92] and [97] use the Hough transform for circle detection. The Hough transform is a widely used tool in the image processing field for detecting circular contours. It is applied to an edge map of the initial image, where the gradient is applied horizontally in the zones of the eyelashes and vertically for detecting the boundary between the sclera and the iris, i.e. emphasizing the shape of the border to be detected. The circles which surround the iris are computed by searching for the maximum values in those edge maps which follow the equation of circumferences. The localization of the outer boundary is performed first, this allows the inner boundary search to be performed in a smaller area as the pupil is always confined within the

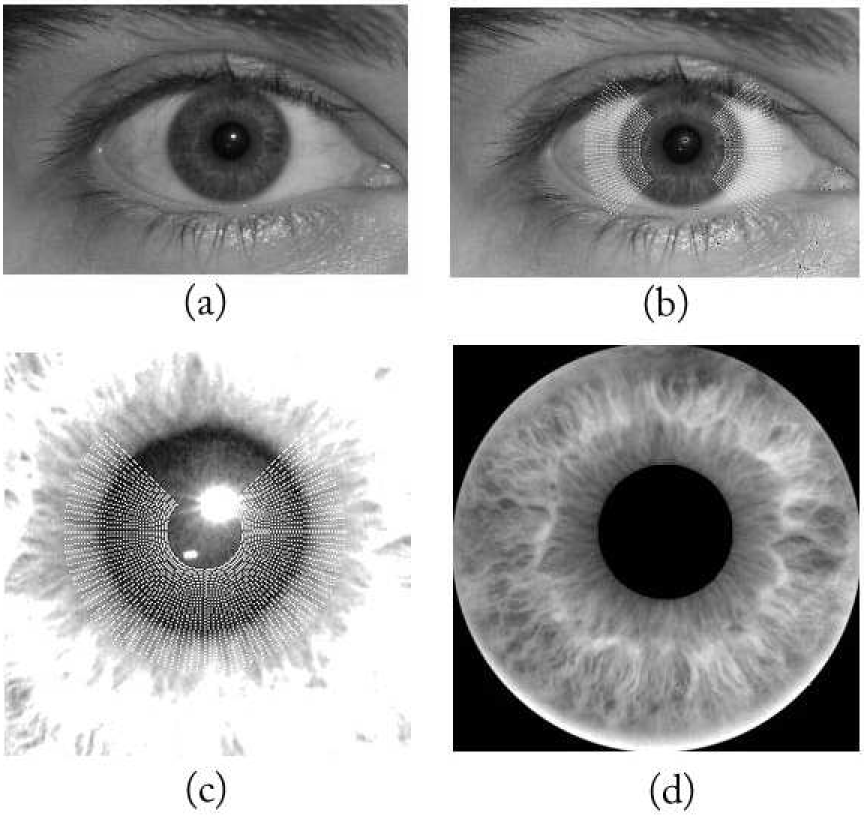


Figure 3.5: Iris Preprocessing by [135]

iris. The main problems associated with this solution is the high computational cost required, as it searches for potential circumferences using a brute force method, and the threshold used to detect the edge map depends significantly on the image. Tisse *et al.*'s proposal [150] combines Daugman's integro-differential operator and the Hough transform. The Hough transform is used to locate the iris within the image, and in the area surrounding this location, the integro-differential operator is applied to find the iris and the pupil. The main problem associated with the methods mentioned above is that they all search for perfect circles within the image. However, as previously mentioned, the iris and the pupils are not perfectly round. In order to make the biometric modality universal, i.e. for all users, the most recent algorithms which have been developed are more flexible with regards previous proposals and therefore, the iris segmentation becomes increasingly viable and a reduced population percentage is rejected. By considering that the pupil and iris are not round, the most common approach is to use an active contour solution. First approaches using this technique have been proposed by Ritter for iris in slit-Lamp images of the Cornea [128]. Following this first approach, other authors have presented different approximations, such as [129] which has performed iris segmentation using geodesic active contours, or Daugman [26], in his latest approach, which relates the active contour using Fourier expansion coefficients, this also provides information on gaze deviation.

3.4.3 Feature extraction block

A wide variety of proposals have been presented for the extraction block feature. The majority of these start by normalizing the iris segmentation. This normalization is necessary to overcome the non-elastic pupil size variations due to light and also possible iris occlusions caused by eyelashes and eyelids.

3.4.3.1 Normalization

The normalization process varies depending on the rest of the processes within this block. The most common and accepted method is to perform coordinate changing, from Cartesian to polar [28]. From this change the resultant images are more compact than previous ones; and depending on the interpolation performed, little data is lost. In this change, it is important to consider that both the pupil and iris are not concentric circles and therefore, the change should take into account this fact:

$$[h]I(x(r,\theta),y(r,\theta)) \rightarrow I(r,\theta) \tag{3.3}$$

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta) \tag{3.4}$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta) \tag{3.5}$$

where (x_p, y_p) and (x_s, y_s) are the coordinates of the pupil and the iris respectively.

Other authors use other normalization techniques which are dependant on the data required by the feature extraction block. Among these other techniques, we highlight the methods proposed by Li Ma *et al.* [91] and Sanchez-Avila *et al.*[135], which only consider a single circular annulus around the pupil, the authors claim that the relevant information is near the pupil and is sufficient for recognition.

3.4.3.2 Feature extraction

After normalization, Daugman studies the phase information by applying different Gabor transforms. Daugman applies both even and odd Gabor filters to obtain the phase information delimited by both the real and imaginary parts of the Gabor filter. With the resulting data, he encodes the information according to the quadrant it belongs to [28]. The codification provides a feature code formed by 2048 bits.

$$\begin{aligned} h_{R_e} = 1 if Re \int_\rho \int_\Phi & e^{-iw(\theta_0-\phi)} e^{(-r_0-\rho)^2/\alpha^2} \\ & e^{\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi \geq 0, \end{aligned} \tag{3.6}$$

$$\begin{aligned} h_{R_e} = 0 if Re \int_\rho \int_\Phi & e^{-iw(\theta_0-\phi)} e^{(-r_0-\rho)^2/\alpha^2} \\ & e^{\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi < 0, \end{aligned} \tag{3.7}$$

$$\begin{aligned} h_{I_m} = 1 if Im \int_\rho \int_\Phi & e^{-iw(\theta_0-\phi)} e^{(-r_0-\rho)^2/\alpha^2} \\ & e^{\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi \geq 0, \end{aligned} \tag{3.8}$$

$$\begin{aligned} h_{I_m} = 1 if Im \int_\rho \int_\Phi & e^{-iw(\theta_0-\phi)} e^{(-r_0-\rho)^2/\alpha^2} \\ & e^{\theta_0-\phi)^2/\beta^2} I(\rho,\phi)\rho d\rho d\phi < 0, \end{aligned} \tag{3.9}$$

$$\tag{3.10}$$

Masek in his master Thesis [97] has performed a similar approach to Daugman by using texture analysis log-gabor filters which eliminate the DC component. In Wildes' approach, the extraction is performed using a Laplacian pyramid of gaussian filters, where several images are obtained of different spatial scales which are then used for posterior comparison [155]. Sanchez-Avila *et al.* have proposed in [135], two different feature extraction approaches: the first uses Gabor filter weighting and the second approach is based on the use of the dyadic wavelet transformation used to detect sharp variations in the iris signature and its zero-crossing representation. Boles *et al.* [13] have also based their proposal on the dyadic wavelet transform, but use a normalized iris as proposed by Daugman, i.e., using a 2-D wavelet transform on the polar scale representation of the iris. Several feature extraction approaches have been presented by the Chinese Academy of Sciences [50]. Li Ma *et al.* have proposed a similar approach to Daugman in [84] and [92], which is based on circular even-symmetric Gabor filters. In [91], their proposal relies on a 1-D dyadic wavelet transformation of the normalized iris presented by Daugman. A completely different proposal has been presented in [146] where the direction of the image gradient vector was studied using gradient vector fields and isotropic derivatives obtained from the Gaussian convolution of the gradient vectors. After that, ordinal measures are used to obtain the feature vector. Their proposal compiles several ordinal measures such as Daugman's, PCA components from Noh [105] or by using a dyadic wavelet, and encode them using a robust coding scheme. Tisse *et al* [150] have based their algorithm on the instantaneous phase and emergent frequency calculated using the Hilbert space and Fourier Transform. Other methods can be found, using Haar wavelets [105], ICA analysis [8], etc.

3.4.4 Comparison Block

The Comparison block depends on the nature of the feature vector obtained from the previous block, and is also named iriscode. The computation of the difference between the sample vector and the template vector varies according to the vector obtained. Wildes' algorithm [155] returns a number of images which represent the iris in different spatial scales. The comparison is performed by computing a normalized correlation between the images obtained and those stored as templates, and therefore, the similarity between images of different spatial scales is what determines the identity of the user. In [84] the iriscode obtained is formed using a vector which results from the integration of different iris sectors where a Gabor filter has been applied, in this case the resulting vector is composed of numerical elements. The dimension of this vector is then reduced using the Fisher linear discriminant. The matching, in this case, relies on a nearest centre classifier based on the mean value obtained from several distances. Several matching algorithms have been studied in [135]to construct the binary code obtained from the zero-crossing representation of a dyadic wavelet transformation. The comparison algorithms studied are the Euclidean distance, the Hamming distance and the Zero-crossing distance. The most commonly used matching technique is performed using the Hamming Distance. Where this method measures the difference/similarity of two binary format vectors. Most of the algorithms which have been presented return a binary vector from the feature extraction block which results from the phase information [28], [97], [150] from the zero-crossing situation [135],[91],[13] . This distance is described by the following formula 3.11:

$$D = \frac{1}{L}\sum_{k=1}^{n}(p_k \otimes y_k) \tag{3.11}$$

Where L is the vector length, p_k and y_k the k-th component of the template and sample vector. If the distance obtained is below a predefined threshold, the sample being identified is considered to belong to the user whose template is being studied. The threshold value is typically determined empirically. Many researchers use this formula or propose minimal modifications to increase the separation between inter-class and intra-class distances for massive identification applications [25],[61]. The modification proposed by Daugman in [25] is shown in 3.13. This modification considers two masks which delimit the bits being compared, because in most cases the iris is partially occluded and therefore, not all the bits from the IrisCode are useful:

$$HD_{norm}(A,B) = 0.5 - (0.5 - HD_{raw})\sqrt{\frac{n}{911}} \tag{3.12}$$

$$HD_{raw}(A,B) = \frac{\|(Code_{A\_Rotated} \otimes Code_B) \cap Mask_{A\_Rotated} \cap Mask_B\|}{n} \tag{3.13}$$

$$n = \|Mask_{A\_Rotated} \cap Mask_B\| \tag{3.14}$$

where $mask_A$ and $mask_B$ are the masks obtained during the pre-processing and the feature extraction block, locating which bits of the feature vector to be considered for measuring differences and which bits come from the Iris zones occluded by eye lashes or eyelids. Factor 911 is empirically obtained as the mean number of bits measured and $\cap$ represents a logic AND operation.

This formula has been used in [25], where the author has demonstrated the feasibility of the algorithm by performing 200 billion comparisons.

3.4.5 Algorithm's comparison

The performance results from different algorithms varies significantly from one algorithm to another, but always claim to have less than 1% of EER. However, it proves difficult to compare them as different databases have been used. In 2005, the National Institute of Standards and Technology presented the first challenge for iris biometric algorithms. This challenge was the first time different algorithms could be compared using the same database. Most of the algorithms which participated in this event were developed in different universities and several commercial companies. The results of this challenge can be seen in fig 3.6.

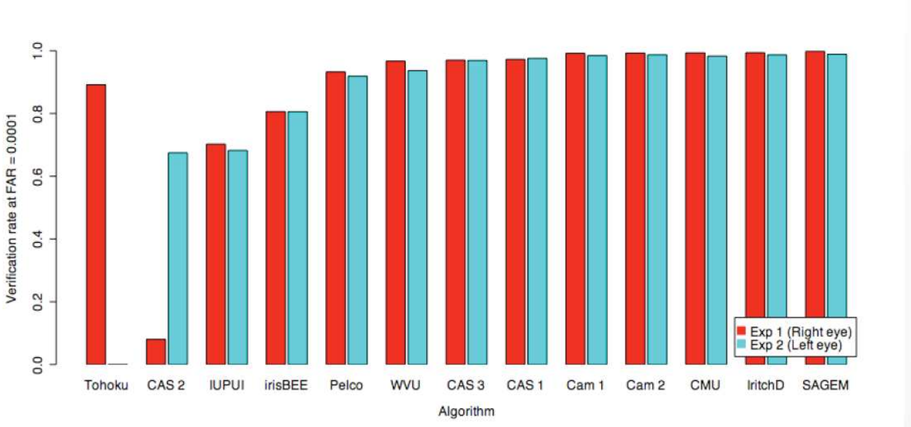


Figure 3.6: Bar plot performance results for fully automatic iris systems with FAR=0.0001 [108]

The previously mentioned challenge was also performed the following year (2006). Two years later, the NIST proposed a study of how iris algorithms accuracy are influenced by compressing the initial images and by only studying the polar representation of them. The results from this study were not available during the confection of this Thesis. However, as Daugman has performed a similar analysis in [27], the decrease in accuracy depends on the algorithm and of course, on the compression level of the initial images. Fortunately, the results provided in this paper only show a small decrease in accuracy (less than 0.02%).

3.5 Base-Line algorithm

The work presented in this Thesis is not focused on the development of an algorithm, although an in depth knowledge of their workings is required to optimize the implementation. The algorithm used in this work has been developed and presented by Sanchez-Avila and Sanchez-Reillo in [135], although some modifications have been carried out to improve the performance of the results. In this section, we provide a detailed description of this algorithm so as to aid in the comprehension of the following chapters.

3.5.1 Pre-processing

The initial pre-processing module proposed in [135] relies on detecting the maximum difference between intervals of bits (Fig. 3.5). However, results obtained with this method when using infrared images were quite poor and do not satisfy the requirements for the recognition purposes. As a result, several modifications have been made to achieve a more suitable algorithm for this type of image. Fig. 3.7 shows the scheme followed in the new pre-processing algorithm.

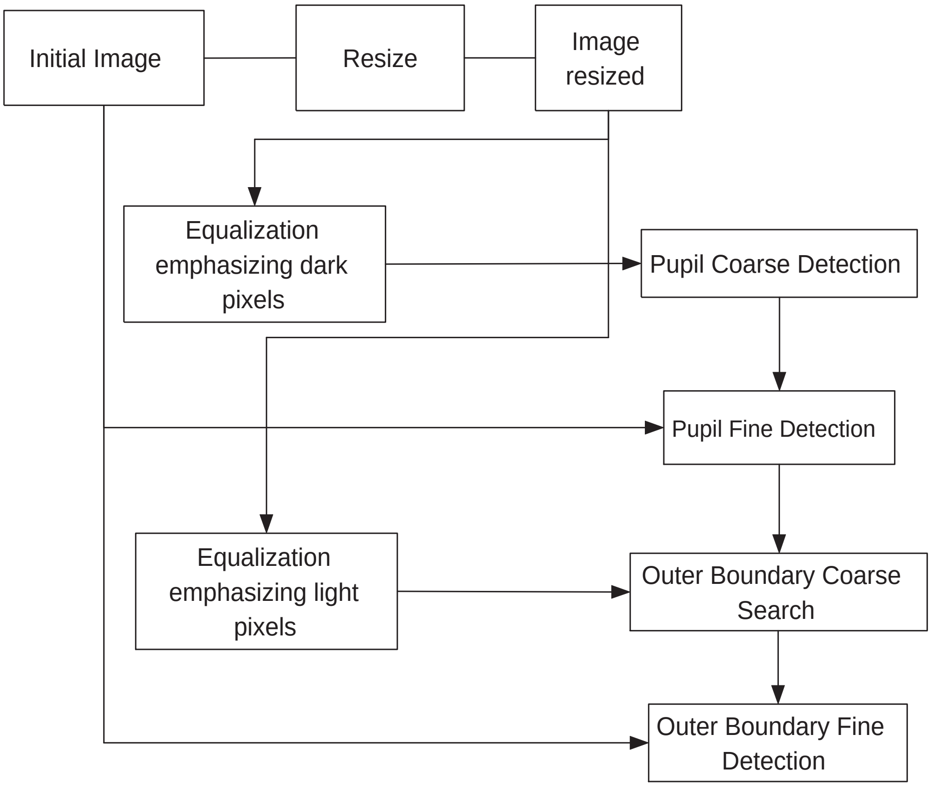

Initial Image
Resize
Image resized
Equalization emphasizing dark pixels
Pupil Coarse Detection
Pupil Fine Detection
Equalization emphasizing light pixels
Outer Boundary Coarse Search
Outer Boundary Fine Detection

Figure 3.7: Pre-processing block of the base-lined algorithm

The pre-processing now relies on the morphological properties of the image: two sets of pixels are considered, those delimited by the pupil and those which belong to the iris, the first set being located inside the second. The approach used for both sets has been two non-concentric circles. This approximation will lead to errors in pre-processing, which the normalization in the feature extraction block will try to overcome. In extracting and delimiting these two pixel sets from infrared images, we should also consider that:

- The pupil clearly shows darker pixels than the iris, however, we should also consider that this property also appears in eyelashes and make-up zones.

- The iris outer boundary is much more complicated to detect for two reasons:
  - The difference between pixels of the iris and the sclera is not as noticeable as the pupil case. The sclera presents lighter values than the pupil, but most of the time, this difference can only be observed in the lateral borders of the iris, but not in the upper or lower border.
  - The iris is mostly occluded by eyelids or eyelashes in the upper and lower border.

The pre-processing starts with pupil detection as it is more straightforward. Before starting with this detection, the image is resized, making it smaller, so as to save on computational time and resources for the image tracking. The first step taken when detecting the boundary of the pupil is to emphasize it. By considering the dark part of the pixels, an equalization is performed which is based on the histogram results, i.e. computing which values are considered to be black pixels, increasing the contrast between these and the rest. Once the small image is equalized, then two parallel searches are performed to find the largest block of these dark pixels in the image. One search is performed in the horizontal direction and the second in the vertical. In both search schemes, potential reflections from flashes are avoided by considering possible groups of light pixels located within a dark block.

Once the image is fully scanned in both directions, the values for the maximum block in both are obtained. With these two values, the image is scanned again in the opposite direction, i.e. after checking the horizontal direction and with the new values obtained, the scan is performed in the vertical direction and vice versa. This second search helps to delimit the pupil area in both cases. After the second scanning is finished, the values obtained from both branches of the algorithm are combined to determine estimated pupil values. These values are those which refer to the large diameter found in both branches, the possible centre coordinates are calculated from the mean of both values.

The pupil fine search is very similar to the one which is performed later to obtain the outer boundary. In this case, we only attempt to consider the lateral cones drawn from the centre coordinates, fixed angle and a smaller range of radii. For this fine search, the approximation initially proposed by Sanchez-Avila *et al*, is considered by checking the lines for the radials from the previous centre. This search is performed on the full size image as we expect more accurate results. The sequence followed to detect the outer boundary is similar to the one presented above for the pupil, the main difference here is that both searches are performed following radial search directions in only two lateral cones of the pupil.

3.5.2 Feature extraction

The feature extraction method chosen has been proposed by Sanchez-Avila *et al.* in [135]. In this algorithm, the iris is normalized to a virtual annulus region around the pupil, where this is known as the iris signature.

$$(x, y) = (x_0 + \Delta r \cos(\Delta\theta), y_0 + \Delta r \sin(\Delta\theta)) \tag{3.15}$$

$$\Delta r = \frac{r_e - r_i}{4} + r_i \qquad \Delta\theta \epsilon (0, 2\pi)$$

Where r_e is the iris radius, r_i the pupil radius and (x_0, y_0) are the coordinates of the centre of the pupil.

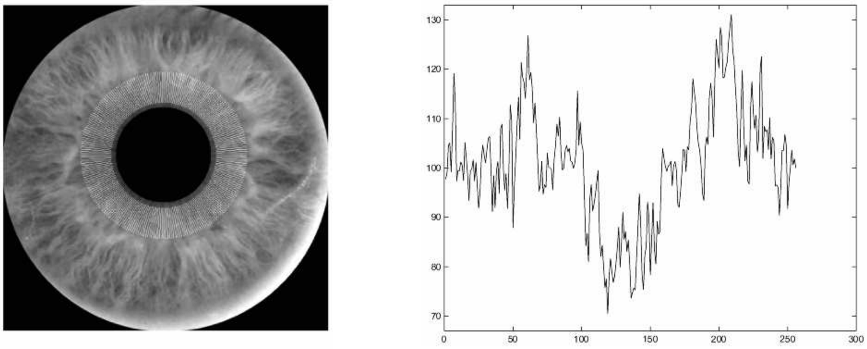


Figure 3.8: Iris signature computation

When computing the annulus region it is verified if the pixels which are considered to belong to it, also belong to the iris itself and do not fall outside the outer boundary. Extracting the features from the iris signatures and representing them is performed using the discrete dyadic wavelet transform [94]; this is followed by its zero crossing representation to obtain a set of "detailed components" from different sizes, considering its different scales.

Mallat's Fast Wavelet Transform (FWT) approach has been considered for computing the wavelet transformation [93].

In this transform, the input vector is processed using a low and a high pass filter. The resulting vector from the high-pass filter is then re-filtered using the same

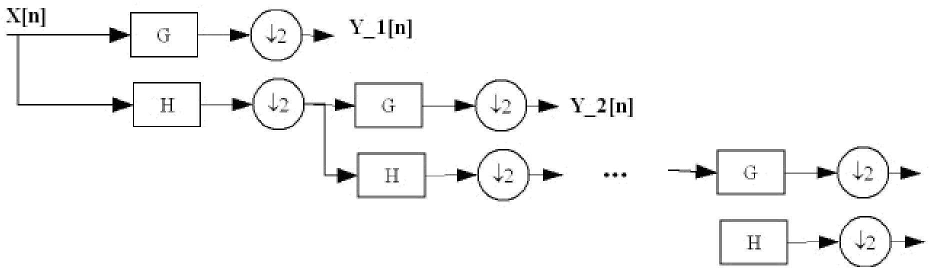


Figure 3.9: Fast wavelet transform

filters, and so on. Each scale of the wavelet is represented by a filtering process. After each filtering, the output vector is down-sampled to maintain the number of bits of the input vector and also, because there is no relevant information in the even elements. In the approach used in this Thesis, the down-sampling stage which follows each filter provides the worst performance results when compared to a zero-insertion in intermediate vectors. Therefore, the resulting vector does not have the same length as the initial vector; its length is the initial one multiplied by the number of scales the transformation performs.

Once the dyadic wavelet is computed, the vector is simplified by using its zero-crossing representation. The zero-crossing representation converts the vector into a binary representation where '1' represents a positive value, and '0' represents a negative value. The resulting vector points out the sharp variations of the iris signature for different scales 3.10.

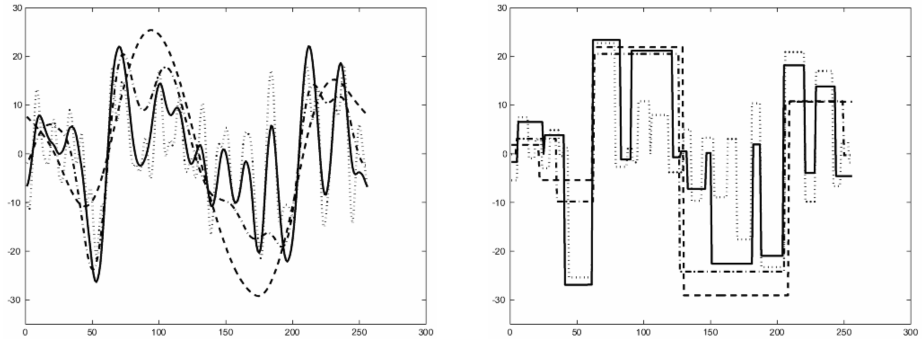


Figure 3.10: Fast wavelet transform and its zero-crossing representation

3.5.3 Comparison

The iriscode obtained from this feature extraction method is a binary code formed from 256 elements⊗n, where n is the number of scales considered. Sanchez-Avila *et* *al* have tested different comparison algorithms using these vectors in [135], and have obtained the best results when using the Hamming Distance.

This distance measurement provides the differences between the sample and template vector. As it can be observed, when the distance is close to 0, the sample is considered to belong to the user whose temple is being examined. On the other hand, if both vectors are very different, where this is translated as a very large difference value, the template is thus considered not to belong to the same user.

Sanchez-Avila *et al.* have presented the performance results for this algorithm considering their own database in [135]. The results obtained have shown an EER Rate of 0.4%. The database they have used is based on photographs which contains iris registers from 20 people and almost 10 samples per user. The quality of the initial images are quite high and show clear iris details. However, when applying the algorithm to any of the previously mentioned databases, the results obtained were observed to be quite different. In this Thesis, we have worked with the ICE2005 database. This database, as opposed to the that used by Sanchez-Avila *et al.*, provides images of different quality, also in many cases the iris is occluded by eyelashes, the user is looking in a direction other than that of the sensor, make-up, etc. As a result the performance results are a lot less efficient than with the initial database. The results obtained are summarized in the following figures:

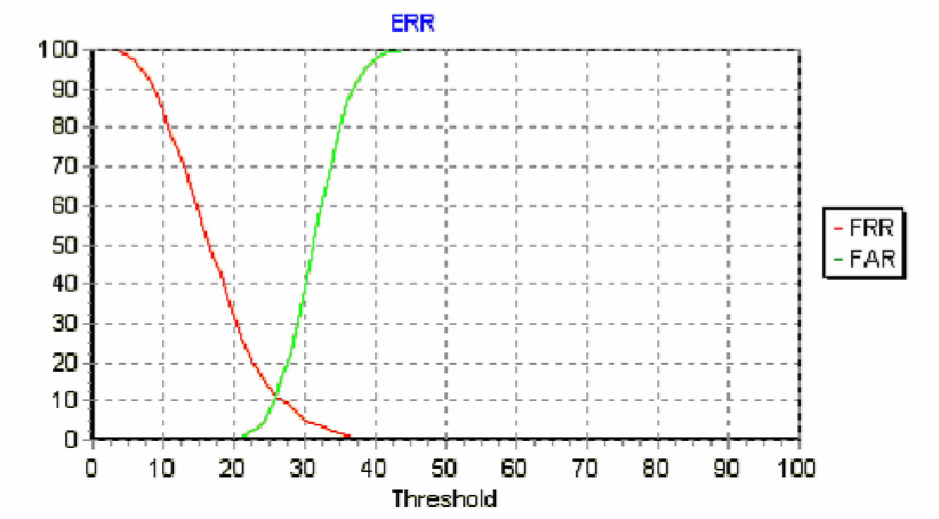


Figure 3.11: FRR and FAR depending on the threshold variations

As it can be seen, from using this database the EER increases to 12%. This increase is dependant on the errors provoked by the number of scales used in the dyadic wavelet transformation, the quality of the images, and the pre-processing block. As previously

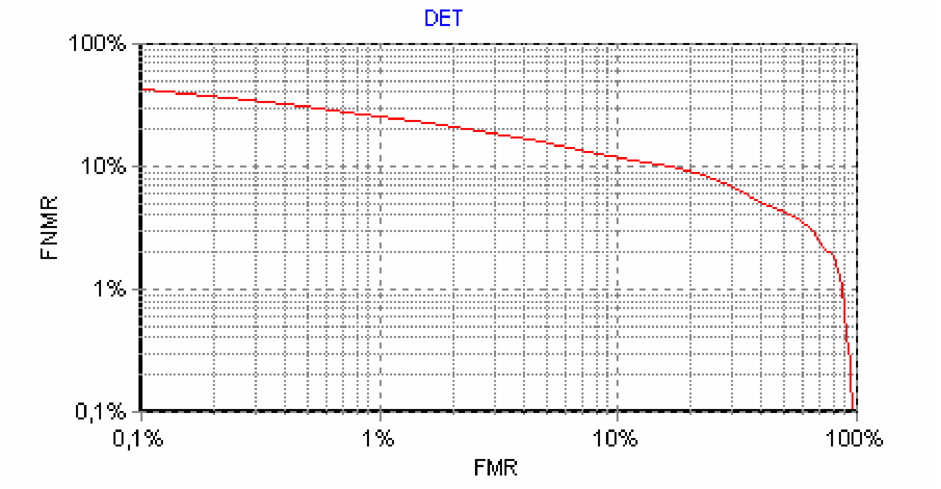


Figure 3.12: DET curve obtained for algorithm implemented

mentioned, improving the algorithm goes beyond the scope of this Thesis, however, we consider it necessary to make modifications to reduce the error rate. Continuing work on the effectiveness of the algorithm is a possible area for future work in biometrics.

3.6 Interoperability

In a society where more remote services are provided each day and where information is shared from different parts of the world, biometric systems are a clear exception. Biometric systems do not provide any of the information they store or manage as this is considered private and obviously they do not provide any information of the algorithm used. However, now that the number of alliances and merging of companies and governments is increasing, the requirement for shared information is becoming an important issue. Currently, the necessity for interconnect systems directly affects the information shared among them. This demand is leading ISO to work on a new standard for inter-system information transfer. Such a standard [52] is undergoing serious discussions among commission members, this indicates that an agreement is far from being reached. In this section, a description of the work developed for this purpose in order to show the possibility of future interconnection between different systems and therefore, the use of different algorithms.

Considering data management, it would be desirable to share feature vectors among systems. However, each system relies on different algorithms and as previously mentioned, the comparison algorithm is pointless if the vectors to be compared have been computed by different systems. For this reason, in Iris Biometrics the data exchange should be carried out in raw format, i.e. images.

In order to reduce the amount of data to be transmitted, several proposals have been made:

- Image Representations:

  Rectilinear representations. This representation is the most commonly used image representation, it is also known as raw image or Cartesian representation:

  - Strict cropping of iris boundaries: the image is pre-processed and its size is reduced to $(2r)^2$, where r is the estimated radius of the iris. This radius should also be provided along with the image as it is necessary for further processing.
  - Un-segmented cropping: due to the difficulties in finding the iris in the image, un-segmented cropping proposes the use of a bigger radius, always making sure that the iris is clearly included in the image, i.e. the size of the image would be $(2(r+u))^2$, where u is the number of uncertainty pixels.

  Polar representation. Considering the normalization methods presented in the feature extraction section of this chapter, the iris can be represented using polar coordinates. Initially polar representation was accepted by the ISO commission, but further analysis determined this method nonviable and it was rejected due to the lack of information and the impossibility of completely reconstructing an image.

  - Direct polar representation: although the pupil and the iris are not perfectly round, they can be approximated by circles. The direct polar representation takes points from circumferences of radius 0.8p to those of radius 1.15i, where p is the estimated pupil radius and i the iris radius. 256 angles have been considered.
  - Representation with bilinear interpolation: the intensity of each polar image sample is computed as the bilinear interpolation of the four closest pixels calculated from the rectilinear input image. When using this representation, the image obtained shows a blurring effect due to the interpolation process.

  - Radial sub-sampling: as the most relevant information for recognition purposes is located near the pupil, it is expected that modifying the radial sample rate influences the resulting size of the image but does not affect the performance results.
  - Angular sub-sampling: Similar to the previous method, but here the number of points considered is larger for the inner radius of the iris and smaller near the outer boundary.

- Image Codifications. As in image processing, the iris image can be subjected to any coding which reduces its size by applying an encoding algorithm to the raw data. The standard algorithms considered are JPEG and JPEG 2000 where lossless compression is recommended.

In [121], [27], [141] and [149] several tests have been performed to check the influence of these compression algorithms on the performance results. The results obtained have shown that the performance may be affected by the compression, but the data gathered is still useable for recognition applications, where the EER is always below 0.01% for several algorithms studied.

In this chapter, we have presented a brief summary of previous work carried out in the area of Iris Biometrics, where the algorithm that will be used as part of the work developed in this Thesis has been highlighted. All the proposals presented have shown the reliability of the iris for recognition purposes and it is becoming one of the most promising techniques with one of the lowest error rates. The use of this modality is increasing every day, and it is now commonly used for access control to several big facilities such as border controls, in several airports and big buildings. However, although these algorithms appear to work quite well, several factors should be studied further, such as the devices required to perform the processes or the security aspects of the system processes.

Chapter 4

Biometrics security

Biometrics, due to its inherent link with personal data, requires high levels of security. Therefore, countermeasures must be taken to protect this data, during both storage and information exchange. In contrast with other identification systems, Biometrics is more sensitive to attacks; even though these attacks are more complex and difficult to execute. The fact that these systems rely on a trait of the user and not on something the user possesses, clearly decreases the probability of ID theft and ID guessing, however this problem is not completely eliminated. Unfortunately, any successful attack on a recognition system leads to irreparable consequences both to the users and the system itself. An intruder may obtain access to restricted and sensitive information, modify it; or even, commit a crime and incriminate an innocent person.

Besides all the problems associated with a non-desired access, another important consequence is the exposed situation to which the system is left. A breach in an identification system requires changes and modifications to avoid future intrusions. In the case of conventional identification systems, these changes are simply a password or token change. However, in Biometrics it is not so straightforward; in fact, a security breach would require changes to the biometric trait being measured. Therefore, if a non-authorized user attempts to access the system with an authorized fingerprint, this fingerprint should be removed from the system and the authorized fingerprint holder should provide another print, i.e. from a different finger. Thus, the number of possible changes is limited by the trait used: the fingerprint modality allows 9 possible changes, Iris Biometrics only allow 1 but face recognition systems do not offer any possible changes. Therefore, an exposed biometric system can even lead to a general change in the infrastructure, with its consequent cost and nuisances.

As the general framework where biometric systems work can not be avoided, in this chapter we will discuss the security requirements these system should have by considering:

1. what are the possible causes of a system failure related to,
2. what kind of attacks should be considered, and finally,
3. what solutions are being proposed and developed.

4.1 Security analysis of biometric systems

The following fishbone diagram has been constructed based on a study of the possible causes for biometric system failure due to security loopholes. In this diagram, all the causes for the security breaches which are discussed in previous standards and papers [71], [138], [122] and [1] are examined and classified depending on their nature. As the diagram shows, the major areas of risk causes can be divided into five groups:

- Attacks referred to as **concealing attacks**, which are characterized by the use of an external device to simulate the biometric characteristic examined by the system.
- All possible **leakage and alteration** of biometric data which is intimately related to authentication attacks. In all these types of attacks, the intruder penetrates the system to obtain information for future use.
- **Administrative tasks**, such as enrolment or inappropriate use of top-level access permits are other weak points for potential attacks by an intruder, due to possible malicious modifications in the data and parameters stored in the system.
- **Infrastructure** vulnerable points are those which are derived from intrusion in the system hardware, violating privacy or introducing bad software between data processes. Countermeasures in this field as well as those for biometric data leakage are well related; however, in this case, not only biometric data but also keys, personal and management data must also be protected.
- Finally those related to the **intrinsic** nature of Biometrics. The fact that Biometrics relies on recognizing people by means of a physical trait implies possible misidentifications due to errors in pattern recognition and similarities among different users traits.

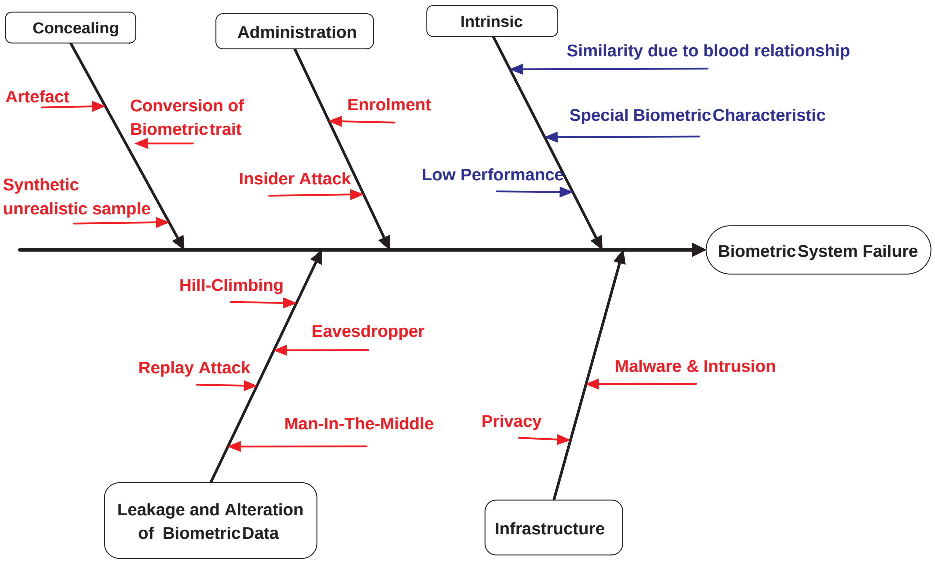


Figure 4.1: Fishbone diagram of possible causes for biometric security failure

Another commonly used method to classify these attacks, other than by causes, is by considering the intruder effort required to breach the system. This classification is remarkable as it allows us to quantify the intruder's knowledge of the system and their intentions. Considering this classification, the attacks can be divided into two main groups [71]:

- Zero-effort, pointed out in blue in Fig. 4.1.
- Adversary attacks, emphasized in red in Fig. 4.1.

4.1.1 Zero-effort attacks

These attacks are accidental, mainly carried out by opportunistic intruders whose biometric trait is relatively similar to a legitimately enrolled individual, thus resulting in a false match. These attacks are intrinsic due to the fact that Biometrics relies on pattern recognition algorithms, and this attack is derived from the probability of observing a degree of similarity between references originating from different sources. Different attacks can be considered within this group [1]:

4.1.1.1 Poor performance of the biometric system

As already mentioned, biometric systems rely on a statistical process which is subject to errors that are expressed in terms of error rates, i.e. false acceptance and false rejection rates. These and other performance parameters have serious implications on the reliability of the security provided by the biometric system. Error rates are usually quantified during the evaluation process by the manufacturer. Although several considerations are made:

- Is the testing environment similar to the place the system is going to be employed?
- What are the optimum conditions for proper system performance?
- Is the test crew adequate for this purpose?
- How many attempts does the system permit prior to user access rejection?
- Are the decision policies and threshold settings appropriate for the environment where the system is going to be operational?
- What error range is acceptable for system performance?

Obtaining the answers to these questions is beyond the scope of this Thesis. However, parallel work on these issues is being carried out within the authors' research group [44], [45] and [139]. A comprehensive guide can be obtained from the ISO/IEC 19795 series of standards.

4.1.1.2 Similarity due to blood relationship

When the biometric trait is determined, mainly from genetic characteristics, users from the same family tend to have similar traits; this condition creates a potential vulnerability point. The most well-known case occurs for face recognition, where brothers and sisters tend to show similar characteristics and may be mistaken for one another, especially when dealing with siblings and twins. When the trait development relies on a stochastic process, such as fingerprint or iris, no similarities between family members is found, as these traits have been developed randomly.

4.1.1.3 Special biometric characteristics

Some biometric traits may manifest a potential vulnerability when users biometric characteristics lead to significantly high error rate. In this case, it is important to prevent an attacker from identifying which users lead to high error rates, so as not to replay this specific data in the system.

4.1.1.4 Temporal variation

As mentioned in chapter 2, the traits desirable for use in Biometrics should not vary with time. However, in some modalities such permanent conditions are not possible. This is the case, for example, when using face recognition as aging changes features. Users may also change their facial features with make-up, plastic surgery or with complements. Such changes may lead to errors in facial recognition or even worse, permit impersonation.

4.1.2 Adversary attacks

These attacks are characterized by an impostor that deliberately tries to impersonate an authorized user. These attacks are probably more dangerous than the zero-effort attacks because of the intruder's harmful intentions. These attacks can be classified within the following groups:

- Leakage and alteration of biometric data.
- Concealing attacks
- Administration attacks
- Infrastructure

4.1.2.1 Leakage and alteration of biometric data

These attacks are typically performed on authentication protocols, and are significantly important in Biometrics due to the sensitivity of the information being handled. The following attacks have been considered [15], [16]:

Eavesdropper: An eavesdropper attack is based on an attacker observing the authentication protocol and the running processes for later analysis, after which eavesdroppers attempt to pose as users of the system.

Man-in-the-middle (MITM): This occurs when an attacker inserts himself between the customer and the verifier in an authentication exchange. The attacker attempts to authenticate himself by simultaneously posing as the customer to the verifier and as the verifier to the customer.

Replay Attacks: Where the attacker records the data of a successful authentication and later replays this information to attempt a false authentication to the verifier.

Hill-Climbing: Consists of exploiting the evolution of the responses of the biometric system, by slightly modifying the input sample, to obtain a fraudulent input sample that may provide a successful comparison.

These attacks are usually combined: first, eavesdropper or man-in-the-middle techniques are used to obtain biometric data from an authorized individual; then the attacker slightly modifies a sample in order to attack the system [122], [71], [152] and [53]

4.1.2.2 Concealing attacks

We have assigned as concealed attacks all those related to biometric trait fakes. If the biometric trait used by the system is easily concealable the system is much more secure when compared to others that are based on an unconcealed trait. These attacks are commonly seen in spy movies where a fingerprint is obtained from a glass impression or voice recordings are used on voice recognition systems. Thus, the trait becomes a possible source for the vulnerability of the system, as fake traits can easily be generated. For this reason, it is important to make the user aware of the vulnerabilities and show them how to avoid attacks and reduce their impact [1].

A list of several intrusion attempts which may be considered in this field are:

Artefact containing biometric characteristics: An artefact of a biometric characteristic is a non-human object that may be accepted as a biometric characteristic by a biometric system, for example, a photographed iris or a prosthetic latex finger [96]. Several countermeasures have been implemented to reduce the effects of these potential attacks such as aliveness detection methods. These kind of methods, as its name indicates, verify life in the sample studied, and are used to prevent artefacts containing biometric trait images or traits from corpses. To cite examples, in the case of Iris Biometrics, the sensor captures a video of the iris and checks for the aliveness by recording the natural dilation and contractions of the iris [111]; as regards Fingerprint Biometrics, sensors may be used to capture not only pressure but also sweat as a means of detecting life in the sample [142].

Conversion of biometric characteristics: Users of a biometric system may intentionally convert their biometric characteristics into a format that creates a potential vulnerability, thus putting the recognition process at risk. This vulnerability point is possible whenever the biometric trait is behavioural or biological, such as voice or signature where users attempt to imitate other users voices or signatures, or make-up in face recognition systems, as an attacker can change his/her face using make-up or plastic surgery to look similar to an authorized user. These attacks may be deliberate or unintentional. In any case, the potential vulnerability related to them should be taken into account. Countermeasures depend on the trait used in the system. For example, in iris, no effort has been made, as it is not possible to change the iris structure without risking vision impairments [143]. However, for other modalities such as signature, detection using online systems is focused not only on the users strokes but on the pressure or slope.

Synthesised unrealistic biometric samples: A potential vulnerability may also exist if a biometric system accepts synthesized biometric samples that match more than one enrolee. The main difference between these samples and artefacts is that the latter attempts to copy a known biometric characteristic. Synthesised biometric samples may be completely different from normal biometric characteristics, for example, a fingerprint containing an unrealistically large number of minutiae. A common countermeasure for this type of vulnerability is the rejection of a characteristic that matches multiple enrolees or that which demonstrates an unrealistic feature vector. [1].

4.1.2.3 Administration attacks

Abuse of the administrative authority of biometric systems can compromise the integrity of the system in two ways:

Enrolment process: The enrolment process should be thoroughly observed, as two possible vulnerability points may emanate from it:

- An attacker may try to enrol himself/herself into the biometric system using an erroneous ID or by using an artefact; thus, it is highly recommended that the enrolment process be performed in a suitable environment and with all the required mechanisms that assure the security of this process.
- The quality of biometric references may affect the security-related error rates; if a high error rate is achieved with a template, this could lead to intentional impersonation attacks.

Insider Attack: This attack is related to authorized users who have top level access permissions and may change biometric data, i.e. access other users' accounts, etc.

4.1.2.4 Infrastructure

Besides all of the above mentioned attacks, here we highlight two potential areas that are related to the system infrastructure: on one hand, malware and intrusion and on the other, those referring to privacy. These attacks are mainly provoked by failures in the infrastructure itself, by which security loopholes arise from the malicious actions of intruders:

Malware & Intrusion: These attacks are performed by obtaining fraudulent software which is introduced into a claimant's computer, i.e. at the user's computing environment. They vary in their sophistication from simple key loggers to advanced trojan horse programs that take control of the users computer. Malicious code attacks may also be aimed at verifier systems.

Privacy: As biometric systems deal with personal information such as the identity of a user from their biometric data aswell as several private issues, this area is an important issue to be dealt with. Two considerations have been outlined in ISO 19792:

- During the recognition or the evaluation process, the user-related information should be protected to a maximum degree; for example, if the information is stored in a database, this should be highly secured. Therefore, data protecting mechanisms should be as reliable as possible.
- Interoperability among biometric systems should be considered. Thus, a biometric template should be available for use in a system other than that for which it was created. This interoperability has been reflected in several standards obtained for the different modalities, and depends on the consensus reached among different researchers and commercial groups.

4.2 Vulnerabilities associated with biometric systems

All the aforementioned attacks can be translated into vulnerable points within the biometric block diagram, i.e. each of these attacks influences the biometric schedule processes in different ways. Therefore if a particular attack is to be avoided, many processes should be protected. This section presents an approach to possible attacks on a biometric system schedule which has already been described. We will also discuss which of these attacks has the most affect on each specific point and how each one may seriously damage the integrity of the complete systems security, thus questioning the vulnerability of the system when a malicious system access is successful.

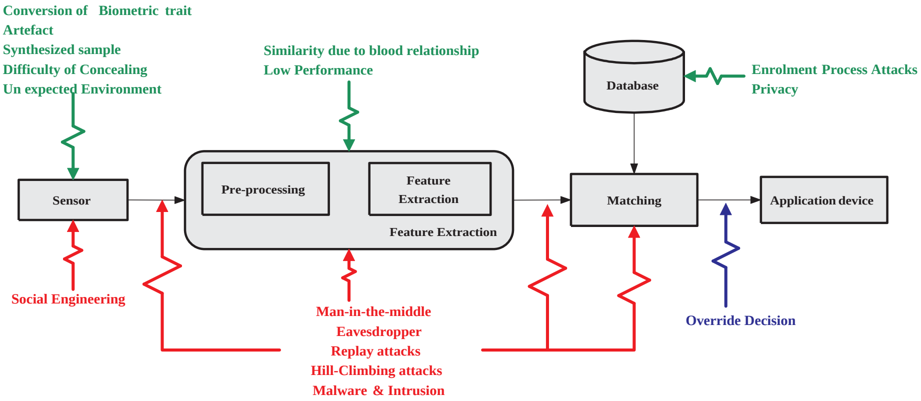


Figure 4.2: Influence of potential attacks on a biometric system

In the above diagram, we have shown the influence of the different aforementioned attacks on each block of the biometric process. In green, the attacks highlighted are related to the intrinsic characteristic of biometric systems and those that are administration-related. In red, are emphasized those which come from authentication protocols and in blue the most sensitive part of the biometric system as regards attacks.

4.2.1 Outside world: intrinsic and environment attacks

The sensor and acquiring process is most sensitive to attacks related with the outside world, such as the case of social engineering, unexpected environment or artefacts. Asking a user to provide unconsciously his/her biometric trait or a replica of the same, such as fake fingerprints, a copy of a signature or a face mask are some of the most common attacks at this point. The possibility of concealing the biometric traits becomes important if the system has to prevent attacks at this point. Moreover, if the sensor performance is sensitive to environmental variations or alterations, i.e. a dusty environment is detrimental to fingerprint sensor performance just as the influence of illumination on some iris or face sensors.

The main efforts in this area are being carried out by sensor manufacturers, who are attempting to increase the security of the sensors, making them less sensitive to environmental alterations or by adding additional features that help detect possible fakes. In the fingerprint field, advanced sensors provide sweat detection in order to reject possible silicon fingerprints; the use of infrared cameras is now quite common to reduce the influence of lighting on face or iris detection, or voice recognition systems which require different texts to be read and so avoid possible voice recordings.

4.2.2 Authentication attacks

Authentication vulnerabilities are more common in the rest of the biometric processes, aside from their own intrinsic attacks. Attempting access to the system by resubmission of old digitally stored biometric signals is critical in all such processes. A potential attacker should listen to the biometric data (eavesdropper, man in the middle attacks) and reuse it in a future access attempt, by simply replicating the information at the adequate point (replay attack) or by modifying it slightly and introducing it into the system (hill-climbing attack). Avoiding these attacks is achieved by making the access to this data much more complex and reducing access by using a smaller number of trusted employees who are in charge of managing system modifications.

Additionally, due to the software properties of such systems, attacks using malware and intrusion should be considered. Trojan horse programs, or similar malware, attempt to access systems and reconfigure them using processes which provide hackers' with positive identifications caused by software modifications. Enrolment process attacks only influence templates stored in the system. Most of the enrolment processes are performed in a secure environment with additional identification security measures. When high levels of security are required, a third party is usually involved, such as government employees or any other party trusted by the system issuer and user. This third party confirms the identity of the user whose enrolment is required.

Finally, the decision made by the biometric system can be overridden, i.e., a cracker can change decisions made by the system. This type of attack is highly sensitive to the security: if the pattern recognition system has high performance and provides excellent security characteristics, but the decision is not protected, all efforts in securing the system are ineffective due to the straightforward process of overriding the result.

4.3 State-of-art of countermeasures for biometric security

The main concern most companies and researchers have is based on adversary attacks. Although zero-effort attacks can still compromise the system, security experts are more concerned with the adversary attacks due to the attacker's malicious intentions. Generally, the circumstances of a zero-effort attack are accidental; however, adversary attacks are performed by prepared attackers with in-depth knowledge of the system. Damage caused by these attackers typically implies more severe consequences than a fortuitous misidentification. Algorithm researchers are currently working on developing improved algorithms with lower error rates. Other areas that researchers are working on are based on finding methods to securing current or future algorithms by adding further security countermeasures to the biometric process. These countermeasures attempt to reduce potential attacks by using a combination of conventional identification systems with Biometrics.

4.3.1 Watermarking

Watermarking techniques are commonly used in ID tokens to complicate potential copies. These techniques basically consist of introducing additional information along with images or data that are not discernible at sight, nevertheless providing a secure method to detect possible changes in the main data. By using this solution, it is possible not only to detect a system intrusion but also attacker copies or forgeries. In Biometrics, several proposals on combining these two ideas have been made to detect possible tampering at the preliminary stages of the biometric process. Recorded images are availed of along with watermarking techniques and introduced within the templates or data. Other proposals in this field are based on the use of barcodes included within fingerprint images [106] or watermarking in voice recordings [43].

4.3.2 Cryptography and Biometrics

Most of the current efforts made by academic researchers falls within this scope. The main objective here is to have biometrics perform user authentication while a generic cryptographic system handles other security-related issues (i.e. secure communication).Thus, if a user accesses a service by using his/her biometric trait, the system will check his/her identity; the biometric decision will release a cryptographic key, which will be used for access to the rest of the system. On the other hand, if the biometric trait presented does not correspond to the user he/she claims to be, the biometric denial will lead to an access rejection. Another typical use for cryptography is in data transmission between terminals and central processors used to protect the biometric data.

However, these systems still have many security problems, such as accessing biometric templates for comparison and its subsequent risk of theft, or those related to authentication protocols such as replay, hill-climbing attacks or even malware and intrusion attacks. For this and other reasons, researchers have changed their efforts towards other solutions, such as Hash functions, non-invertible transformations and biometric keys which combine cryptography and Biometrics to complement each other:

- First approaches have been based on a transformation which uses the biometric feature vector and a key the user should provide (Fig. 4.3). The functions used for transforming the biometric feature vector in combination with the key are called hash functions or non-invertible functions; in such cases, an inverse transformation is almost impossible. These systems, instead of just storing the biometric template, also store the transformation between the biometric template and the user's key. In the identification or verification process, the biometric template is obtained and should be combined with the key, also provided by the user. Both of these information sources provide data which leads to a transformed vector which is compared with the previously stored vector. These systems increase the security level by means of a combination of passwords and biometric systems, as both of them are required. However, they still have a significant problem: the user should remember the key which for higher security levels, should be long and also difficult to guess (but also difficult to remember).

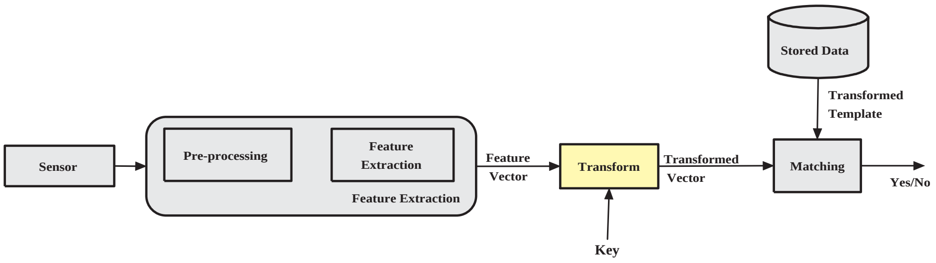


Figure 4.3: Fuzzy-vault scheme

- Systems based on biometric-based keys or fuzzy-vault systems are those which rely on a key obtained from the biometric feature vector. When the user makes several attempts to access the system, the biometric feature vector is used to obtain the cryptographic key, where the user is not required to remember any password. Several approaches [61], [151] have been made in this field, where these have mainly been based on the use of error tolerant representations of biometric vectors. These efforts attempt to overcome the inconveniences and difficulties of these methods, such as finding an encrypted domain where a similar metric for comparison may be used or dealing with comparisons in a transformed domain of unequal transformed vectors. Within these kind of systems, helper data systems may also be included, where during the enrolment process, the feature extraction blocks provide helper data which is stored along with the template. This data can be made public and it is used in later attempts to make an identification decision.

However, these systems do not solve the problem of attacks coming from biometric data leakage during the transformation process, hill climbing, malware and intrusion attacks.

4.3.3 Biometrics and tokens

Another way of securing Biometrics is the integration of Biometrics into token devices. These tokens are called Identification tokens, or ID Tokens. Combining Biometrics and ID tokens is based on the identification of users with conventional ID tokens, such as magnetic cards or keys, where the possession of this device is only by the authorized users. Contrary to biometric and cryptography solutions, major efforts on developing ID tokens has been carried out in industry thus this solution is commonly found commercially. Part of the work developed in this Thesis has been carried out in this field. Chapter 7 will cover this type of device, where all the problems encountered will be described along with token requirements and benefits.

In this first part we have described Biometrics from different points of view. We have made a brief introduction to general Biometrics in the first chapter, considering different aspects of it, from the different architectures it allows to different modalities used for recognition. The second chapter has focused on the modality this Thesis has dealt with, Iris Biometrics. We have described several algorithms used for this purpose and work currently being performed for interoperation among different systems and algorithms. Finally, as regards identification systems, we have described the vulnerabilities and potential attacks biometric systems are under, to understand that even though these systems are more secure than other identification systems, they are still subjected to possible intruders.

Part II

Hardware-Software Co-Design

Chapter 5

Embedded systems

The title of this Thesis is “Hardware/Software Architectures for Iris Biometrics”. Up to this point, we have already discussed how iris recognition systems work and have briefly presented the ID token. However, ”what about hardware/software architectures?” This term makes reference to a methodology used to develop embedded systems. In this chapter, we will discuss different topics related to embedded system designs and their different alternatives. From the architectural point of view, an ID token may be considered as an embedded system, where its main function is to perform the biometric recognition process. For this reason, in this chapter we will describe what an embedded system is, how it works and the different design alternatives. The following chapter will be focused on the design alternative chosen and presented in this Thesis: the hardware/software co-design. Here the advantages and disadvantages of these designs will be highlighted along with the solutions that have been proposed to achieve an efficient design. Finally in this part, we will deal with ID tokens, considering them not only as means of providing identification, but also as a physical device with specific requirements due to the application environment.

5.1 Definition of an embedded system

An exact definition of an embedded system is rather complex due to the large variety of them and the great difference in their designs and shapes. An embedded system is a generalized term for many systems which satisfy all or at least most of the following requirements [10][104]:

- An embedded system is always designed for a specific purpose, although in some cases it can be used for other tasks that it has not been designed for. For example, embedded systems are designed to send infrared signals when a user presses a button on a remote control or to perform floating point operations. In cases where the system is used for tasks other than that for which it has been designed, the performance is significantly reduced or in some cases null. A functionality change in an embedded system has to be done by an expert user, this is because it requires knowledge of the devices internal architecture. As opposed to embedded systems, PCs are designed to perform several tasks and are therefore considered as a general-purpose device.

- Due to their performance restrictions, an embedded system is typically small in dimensions. The size of an electronic device is highly influenced by the number of peripherals and boards required to perform the task it is designed for. In the PC case, several external peripherals are required such as a screen, mouse and keyboard, as well as internal peripherals, i.e. hard disk, video enhancement boards and cooling mechanisms. In embedded systems, only the necessary peripherals are used.

- The cost of these systems is lower than that of a general purpose machine, the cost has as a direct influence on the optimized specific task design.

- These systems usually make use of ROM memories to store their programs and not hard disks or any other big storage systems. ROM memories reduce the storage capability, and therefore, most of the systems use real-time operating systems (RTOS) or an embedded code called firmware, where its size is significantly smaller than general purpose operating systems.

- Due to the application, most of these systems work under real time constraints, such applications range from time sensitive to time critical. For mobile telephones, time is important but not restrictive as opposed to the instantaneous response required from a car airbag control system.

- There is a large variety of applications for these type of systems, from household appliances to large engine controls. Depending on the functionality, power restrictions can be applied, i.e. use of batteries or power supply from a separate system.

- They may also form part of a larger system. Thus, an embedded system may be a stand-alone device or a co-processor, such as game consoles versus the remote control reception system of any house appliance. This feature is strongly related

> to the previously mentioned point, when the system is stand-alone, it is typically battery-powered as opposed to those which form part of a larger system where the power for the embedded system is supplied by an external supply or separate system.

The term embedded system is quite ample, so too is its deployment. Many systems can be found on today's markets, from different peripherals which control big systems to small amplifier systems found in hearing aids. The large deployment of these systems has been motivated by the reduction in the area of the hardware required to implement complex systems: The increase in the number of transistors per hardware area has helped to develop more complex systems within a reduced space which are more specialized in performing determined tasks. For this reason, it is possible to find large quantities of embedded systems on the market and this number is constantly increasing as new embedded systems are available which provide increased performance within a smaller space.

5.2 General architecture of an embedded system

As previously mentioned, it is difficult to present a general architecture for an embedded system, this is because each system is designed for a specific purpose. Therefore, the architecture of each system is unique and optimized for the specific task to be performed. However, we have developed a generalized structure, which provides a general architecture under which most of these systems can be classified, where the embedded systems are only subject to various modifications depending on the performance objectives [60], [104].

As Fig. 5.1 shows, most embedded systems are formed by a central processor unit, one or more peripherals and several memories which are required for different tasks. All these elements are connected using two main paths or buses, one which controls the processes and the second, the data transferral. The protocols these buses follow depends on each device and the peripherals connected to them.

The main processor may be a microprocessor (or a microcontroller), a specific chip, an FPGA, etc. These devices are easily configurable and are capable of performing almost any program. Generally, this main processor is in charge of controlling the overall performance of the system, although when necessary, it can also perform other tasks. For this reason, it is usually connected to both busses so that it can access peripherals and memories depending on the task required.

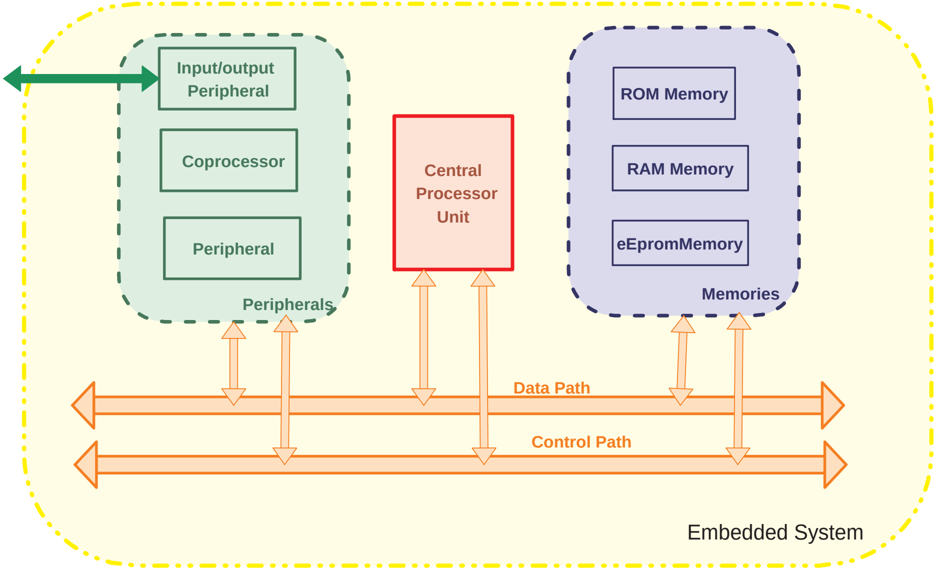


Figure 5.1: Embedded system general architecture

Peripherals are only able to perform a single function. In an embedded system, peripherals have a more extended meaning than those commonly refered to as computer peripherals, i.e. the mouse and keyboard. In an embedded system, chips can be found which are used to convert analogue data to digital data or others used for temporal purposes.

Among all the possible peripherals, here we highlight two different types because of their relevance in the system:

- Input/output peripherals: these are used to interact with the outer world and therefore, are in charge of acquiring data for the system or receiving orders from a user or other devices. Typical input/output peripherals are USB ports, screens, keyboards, etc.

- The so-called co-processors: A co-processor is a dedicated processor which is designed to perform a single function, this processor is implemented to perform the required task faster than any traditional microprocessor. In an embedded system, one or several co-processors are developed according to the systems requirements. The platforms used to implement these devices also varies accordingly, and range from FPGAs to devices based on microcontrollers, such as those dedicated to floating point operations or video management. Many co-processors, as their application are widely employed, are commercialized, however several must be specifically designed. This particular case is studied in the section dealing with hardware/software co-design, which is introduced at the end of this chapter and presented in detail in the following chapter.

Other important blocks of the embedded system are the memory blocks. Memories are always necessary for different purposes, thus the number of memories required varies from one system to another and depends on the application. Commonly used memories are listed below:

- Memory for storing temporary data: RAM memories
- Memory for storing fixed non-volatile data such as programs: ROM memories
- EEprom memory or flash memory may be required to store data which is required to be kept even when the device is switched off, i.e. biometric templates and personal user data. This data should not be stored in the ROM memory, if the case arises where the biometric data is stolen, then this data must be changed.

Memories should be easily accessed by the peripherals and central processor unit. Precisely, such circumstances leads to one of the major problems associated with this type of system, memory access may become a bottleneck if several components try to access the information simultaneously. In this case, a memory arbiter is required to control the access, thus avoiding potential simultaneous access to the memory or even to the same data within the memory.

5.3 Architecture Alternatives

In designing embedded systems, several architectures may be used [104]; this depends on the central element and the scheme to be followed. In this section, we will introduce some of these architectures, pointing out their advantages and disadvantages and also some of their typical applications.

5.3.1 Microprocessor

The most commonly used solution is based on the use of a microprocessor or a microcontroller as the central processor unit [60], [72]. Surrounding this microprocessor several peripherals are required to perform the desired function. These peripherals are different and their communication with the central microprocessor may follow different protocols. The complexity of the peripherals also varies depending on the function the complete system must perform, from straightforward peripherals i.e. analogue-digital converters to more sophisticated peripherals such as USB drivers, which are formed by an additional microprocessor and its related hardware.

This architecture is commonly used and can be found in several applications such as portable game consoles, mobile phones and simpler systems such as the control unit of a washing machine. The specific purpose the system is designed for influences the selection of the microprocessor to be employed: microprocessors used to control processes are generally able to manage 8 bit lenght data; those specialized for Digital Signal Processing (DSP) have a low memory access time and are capable of performing multiply-accumulate operations within a single work cycle.

Microprocessors or microcontrollers require code to be executed. Due to this requirement this type of solution is often referred to as a software design for an embedded system. These solutions perform the code sequentially and therefore, several tasks are not time-optimized or must wait to be performed as the microprocessor is only able to perform one task at a time. The execution program can be divided into any of the following types:

- A real-time operating system, such as Android or Linux Embedded. These operating systems are similar to those used in PC's, however they only include the drivers necessary for the platform being used and the code implemented. These solutions have the capability to load and develop new programs where no knowledge of the internal configuration of the system is required. Their use is increasing everyday and can be found in portable agenda devices (PDAs), mobile phones or mp3 players. The main disadvantage to using these operating systems is related to the storage capacity and the high performance required to make the system work efficiently, which increases the total system cost.

- When the task to be performed is implemented directly on the microprocessor, it is usually referred to as a firmware solution. In this case, improved use of the microprocessor is made as the code is more efficiently optimized for the platform and the task. Firmware code is usually carried out using high or medium level programming languages, such as C language; however, if a more optimized design is desired or the program storage capability is extremely reduced, assembly code should be used. Therefore, using firmware requires specialized designers who are capable of programming microprocessors with high levels of resource optimization.

The main advantages associated with this architecture are those related to the widely extended use of these types of systems. As a result, many efficient drivers and property intellectual processors cores are easily found where manufacturers also provide several libraries required to perform commonly used functions. These libraries, which are provided by manufacturers, are highly optimized for the processor they are designed for. However, in many cases, if they are used along with other processors a poorer performance than expected is provoked.

Another important advantage pertaining to these systems is related to the potential upgrades. When designing these systems, the option to upgrade the firmware is generally available. Currently, it is quite common in electronic devices such as DVD players and mobile phones to update their programs using a firmware upgrade. This solution prevents the system from becoming obsolete and also changes or improves their functionality by means of new firmware, e.g. decoding new video compression formats in DVD players.

These architectures are quite flexible, and at the same time, are relatively easy to work with, this due to all the facilities manufacturers provide, such as programs which make the simulation and debugging more straightforward. All these advantages have prompted the widespread use of this type of architecture; and the design time required is relatively low. The main disadvantage to this type of system is that they are an expensive alternative when compared to other solutions; also these kind of solutions are not as fast as others.

It is also important to highlight systems that use more than one microprocessor, one as a central unit and the rest as co-processors used for different purposes. These are used in high performance systems, where several microprocessors are used for different tasks. For example, in the case of the game console Playstation2, this system is formed using 4 microprocessors, two of which are called vector units and are in charge of graphics acceleration; a separate microprocessor is used for the input/output control and finally, the fourth is used as a central processor unit which controls the complete consoles performance.

5.3.2 Full custom circuits - ASICs

When massive production and high performance is desired, full custom circuits are the best option. As indicated by its name, a full custom circuit is a hardware solution created to perform a specific task, such as measuring a range of data or creating signals according to the system input. Due to this, the design should be carried out by a specialized designer with experience, resulting in the development of a specialized solution.

The main problem with this type of solutions is related to their fixed costs: designing a solution of this type requires not only an experienced designer, which is more expensive than other personnel, but also specific developing tools, longer design time and more complex facilities when compared to other more straightforward architectures.

However, once designed, these systems are cheaper than other solutions with respect to the manufacturing process. The investment can be recovered by the massive production of these systems, as the cost per unit is greatly reduced when compared to microprocessor architectures. At the same time, the hardware area required for these systems is smaller than other solutions which are used to perform the same task, this makes this solution suitable for small devices and helps to reduce the cost per unit.

The upgrading possibility is variable and depends on the hardware developed, but in most cases, it is not possible as the hardware may not be rebuilt. As a result of this, these solutions are considered to be closed designs.

Finally, one major advantages to this type of solution is the time reduction in the performance process. This reduction is a direct consequence of using dedicated hardware, as concurrent processes can be performed simultaneously, no time is wasted in waiting, the hardware used is the most suitable for the tasks developed, etc. Because of the above mentioned reasons, full custom solutions are being used less and less each day, and are only being used in embedded systems where the performance is applicable to different systems and therefore, can be commercialized for different purposes.

5.3.3 Combining solutions

Hardware/software architectures [30],[40]are a half way solution when considering the previously discussed alternatives. These architectures are characterized by the use of both possible solutions in order to obtain the benefits from both types of systems; they use full custom solutions, i.e. dedicated hardware, to perform some tasks and a microprocessor to perform others. By using this combination, the inherent advantages of both systems are obtained: such as reduced time, reduced area and also low power consumption.

However several disadvantages are also inherited: the designer in this case, should be familiar with both solutions; and the expected design time is greater than that when considering only the microprocessor architectures. In spite of this disadvantage, such combinations are commonly used and are being studied for embedded systems.

This solution is directly related to the co-processor approach mentioned in previous sections. When the co-processor design is performed exclusively for the system and using dedicated hardware, we can consider it as a combination solution which is based on hardware and software, where both solutions are combined to obtain a more efficient solution when compared to that when only hardware or software is used.

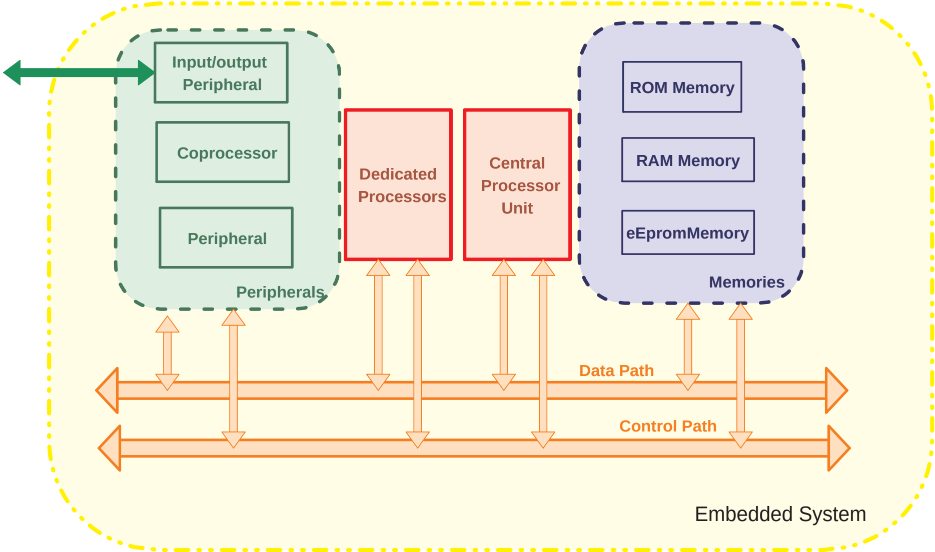


Figure 5.2: Embedded system architecture based on HW/SW co-design

Among all the potential architectures, we would like to highlight the differences between conventional hardware/software solutions and System on Chip (SoC) solutions. In the first case, the hardware is implemented on a dedicated chip and used to perform the task. For example, a high performance analog-digital converter used in a microphone and the microprocessor which manages the information obtained. In this case, the microprocessor and the peripheral are located on different chips. However, due to the transistor integration capacity, many chips which contain both a microprocessor and dedicated hardware are becoming increasingly popular. These chips are a half way solution between the conventional microprocessors and the full custom chip solutions. Among the different commercial alternatives Field Programmable Gate Arrays (FPGA) are currently regarded as the most interesting.

A FPGA is a semiconductor device that can be configured by the customer or designer after manufacturing-hence the name ”field-programmable”. FPGAs are programmed using a logic circuit diagram or source code based on a hardware description language (HDL) to specify how the chip is to operate. The FPGA can be used to implement any logical function that an application-specific integrated circuit (ASIC) can perform, but the ability to update the functionality after manufacturing offers advantages for many applications.

FPGAs contain programmable logic components called ”logic blocks”, and a hierarchy of reconfigurable interconnections that allow the blocks to be ”wired together” - this could be described as a one-chip programmable protoboard. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates such as AND and XOR. In most FPGAs, the logic blocks also include memory elements, these may be simple flip-flops or more complete memory blocks. In the past few years, the trend has been to connect the traditional logic blocks to the embedded microprocessors within the chip. This provides the possibility for the development of combined solutions. These are commonly referred to as System on Chip (SoC) solutions.

As regards the microprocessor used in FPGAs or SoCs, two possibilities may be found: a hard processor, i.e. a processor that is physically embedded in the system, and a soft processor, the latter is implemented using the FPGA logic blocks, providing additional functions if desired by including extra features in the system.

In this Thesis, the last approach has been followed for the biometric task, this is because the potential features which can be obtained using these architectures can help to develop more efficient devices. In the following chapter, we will study, in more detail, how a design following this alternative should be carried out, this has been the methodology used to design and investigate different proposals for each step of the design process. This has also helped to understand the two proposals for embedded systems presented in Part III. One proposal demonstrates the benefits of an embedded system when only considering the time optimization and also considers the importance of an optimized platform design. The second proposal makes use of a hardware/software co-design to develop ID tokens by considering different aspects of the application.

Chapter 6

Hardware/software co-design

Among all the different alternatives existing for the design of embedded systems, the hardware/software co-design has been chosen where the work developed within this field is presented in this Thesis. This architecture is more complex than others and requires knowledge of both hardware and software. In spite of the difficulties that arise, co-design provides several advantages when compared to other alternative architectures:

- Using specific hardware increases the speed of the complete system. However, the lack of floating point operations can reduce the systems performance and some processes may be performed sequentially and thus, the hardware solution may not result in being a relevant time-saver. Aside from the aforementioned issues, different architectures and implementations in terms of the design efficiency is clearly influenced by the designer's experience.

- When only a software solution is used the implementation design and tests are relatively straightforward, however real-time solutions may suffer from time delays. Additionally, using software-based solutions in several embedded systems with high performance requirements increases the system cost, as complex microprocessors are required.

When the computational load of the system is high and it is under real-time restrictions, the alternative of using both approaches becomes popular and effective. By combining both solutions, advantages from both techniques are obtained: the simplicity of using software for sequential processes, and the use of dedicated hardware to perform specific tasks which require high performance. As a result, process acceleration is achieved due to the effective combination of both solutions. However, several parts of the design process become more complicated, requiring additional tedious tests. In the case where both solutions are combined, the designer must have experience in both hardware and software solutions. Not only problems from each platform should be solved but also those related to the connections and communication between them, here several decisions must be made:

- Which parts of the process should be performed using hardware and which using software?
- Which sub-process should be performed first when several of them can be performed at the same time?
- How can we test the complete system?

In this chapter, a description of the hardware/software co-design flow is presented, followed by details of which processes should be followed in the design process (section 6.1). Also in this chapter, the alternatives found from the literal review are presented in section 6.2.1.

6.1 Designing hardware/software embedded systems

The objective here is to answer the aforementioned questions, the hardware/software co-design has been studied in-depth during the last few years by considering the use of both techniques and their implementation as part of a single system, where this provides the advantages pertaining to both methods [40],[30]. Fig 6.1 shows the design process followed for a hardware/software approximation:

As seen in Fig. 6.1, it is necessary to analyze the problem in depth from different perspectives. The first step taken has been to analyze the problem from the point of view of separated hardware and software solutions and creating parallel software and hardware modules for their posterior combination. Once both solutions were implemented on hardware and software and with the data obtained, the problem of how to combine them was then studied. For this purpose the partition problem analyzes different combinations of the possible architectures, where the complete process is split into different blocks where the task using both hardware and software is performed on each one. Once the decision was made on which modules were to be performed in each platform, the integration was then made by following a hardware/software co-simulation. This simulation is important as it determines the feasibility of the solution proposed for the system during early design stages. This simulation aids the designer

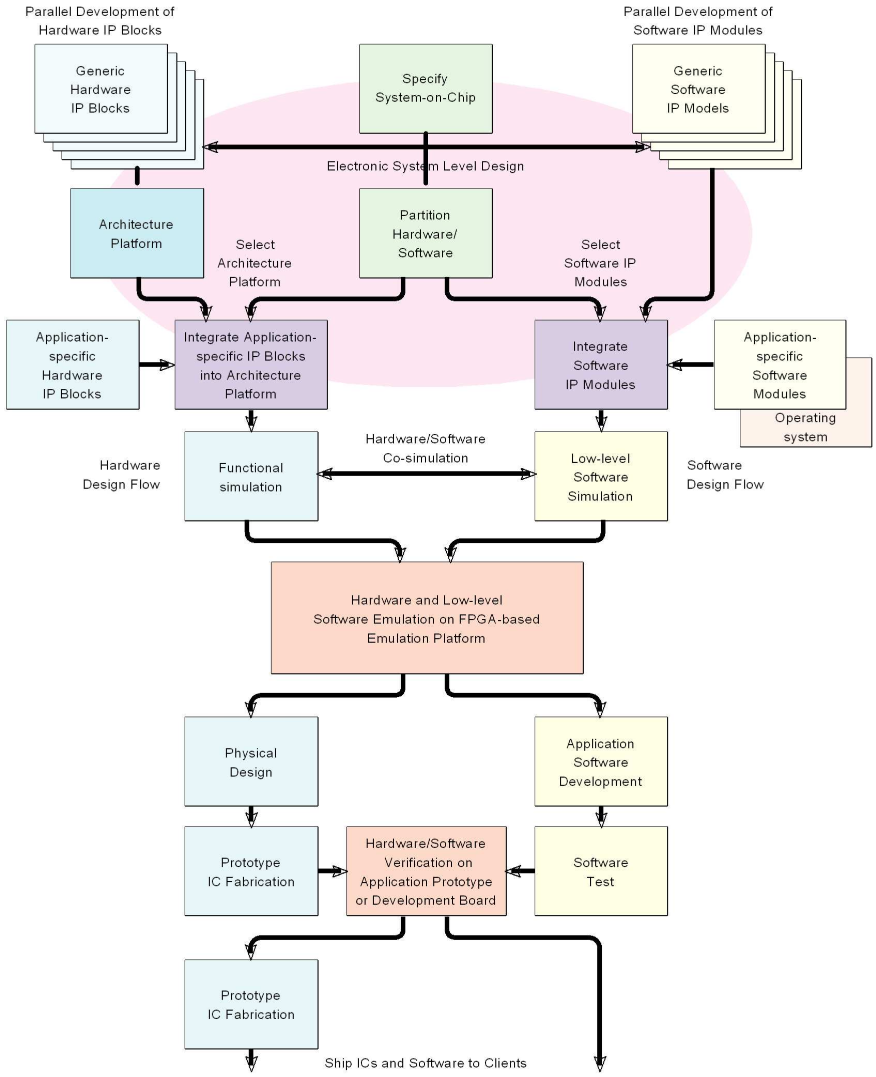

Parallel Development of Hardware IP Blocks
Parallel Development of Software IP Modules
Generic Hardware IP Blocks
Specify System-on-Chip
Generic Software IP Models
Electronic System Level Design
Architecture Platform
Partition Hardware/ Software
Select Architecture Platform
Select Software IP Modules
Application-specific Hardware IP Blocks
Integrate Application-specific IP Blocks into Architecture Platform
Integrate Software IP Modules
Application-specific Software Modules
Operating system
Hardware Design Flow
Functional simulation
Hardware/Software Co-simulation
Low-level Software Simulation
Software Design Flow
Hardware and Low-level Software Emulation on FPGA-based Emulation Platform
Physical Design
Application Software Development
Prototype IC Fabrication
Hardware/Software Verification on Application Prototype or Development Board
Software Test
Prototype IC Fabrication
Ship ICs and Software to Clients

Figure 6.1: Hardware/software co-design flow chart [33]

to quantify the initial performance times, hardware area and the general system performance. If the results obtained from this simulation are not as expected, the designer is able to retrace his/her steps and to perform again any other desired implementation or another partition solution. This simulation is usually performed on a platform which is capable of combining both solutions, such as FPGAs, where microprocessors and co-processors can be implemented on the same device where debugging tools are provided by manufacturers for this purpose.

Once the results obtained achieve the desired level, the designer continues to work on the development of the final solution, i.e. the physical design. This design becomes an initial prototype for the embedded system, where again, tests should be performed to verify the functionality, times, etc. In this prototype it is important to focus on the communication between the software and the hardware modules and among the different peripherals required by the system. If this prototype does not fulfil the initial requirements, the physical design must be changed as many times as necessary until the embedded system fulfils the desired requirements.

From the previous description it can be seen that the design process of an embedded system is a difficult and often recursive process. Some processes of this design are highly dependent on the system requirements and its expected functionality. Several authors have studied this design flow proposing the use of several algorithms for different design and development stages, such as [30]:

Partitioning: is the act of dividing the processes so that each one performs a particular process within the embedded system. The partition problem considers the different possible solutions which depend on the environment where the final embedded system is to work, and the characteristics of each particular module [7], [5].

Scheduling processes: this is done to decide which processes are to be executed and in which chronological order, and to control the access to shared system resources. Such order is partly established by the dataflow process or control dependency.

In this Thesis several proposals have been made and presented here on the combination of hardware/software co-designs into a single biometric embedded system. The proposals have been based on the general theory of Biometrics which were presented in the theory of hardware/software co-design. To understand the proposals presented and the conclusions observed, it has been considered of vital importance to introduce previous works carried out in the area of hardware/software co-design.

6.2 Partitioning and scheduling

Partitioning is responsible for determining which of the processes should be performed using hardware and which using software. This design stage can be carried out by designers who have previous experience using cores provided by manufacturers. This process may also be carried out automatically using computer-aided design tools. Most experimented designers have based their designs on detecting which processes require the most amount of time or designed in terms of those which have a heavier computational load [157], [89], [49], [48], [47], [37] and [156]. By using computer-aided tools, the partition problem can go further; making some processes which fulfil the requirements of computational load or processing time to be performed using hardware rather than software for reasons such as the area required.

Another important fact to be considered during partitioning is the re-usage of previous designs or implementations, such as the intellectual property cores which can be optimized to perform several tasks or cores provided by manufacturers which make efficient use of the platforms resources. Partitioning combines previous designs where the resulting structure is more modular and also considers certain aspects that the designer is not able to quantify until the complete implementation has been carried out.

To include these other factors, the partition problem is generally presented as a function which summarizes the requirements to be satisfied: **the cost function**. This function measures the cost attributed to the different nodes and elements which form the system by considering different parameters. In applying this function, the operation of the embedded system can be studied using its graph diagram. The graph diagram represents the different processes the complete system should perform sequentially and also indicates the sequence these processes follow. The partition function should be

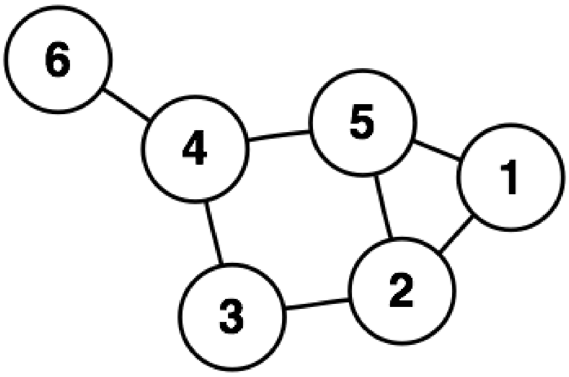


Figure 6.2: Graph Example

applied by considering each of the nodes and their related edges in terms of data graphs that appear in the system. The partition problem may be as intensive as desired, i.e. if an in-depth study is desired, the graph will indicate more specific nodes. The more specific the nodes are drawn, the more nodes there are to be considered; and also, the partitioning should be carried out at a more optimized level. However, this depth is limited by the time and data to be transferred, a trade-off must be found between the time wasted in each transmission and the processes to be performed. A more general form of the cost function is as follows:

$$Cost = \sum_n A_n X_n + \sum_n B_n Y_n + \sum_n C_n Z_n \tag{6.1}$$

The partition problem attempts to discover which of the configurations minimizes this equation, This is done by examining different configurations, and verifying the values to obtain the minimum cost. Each solution is represented by a configuration where some nodes are performed on hardware and some on software. Depending on the implementation desired for each node, the area occupied, processing time and several other parameters will vary. In this formula, each of the summations represents each parameter to be considered and each of its addends, and node contributions to this parameter according to its configuration, i.e. if it is performed using hardware or software. The coefficients applied to each summation (A_n, B_n,C_n) allow the designer to highlight the terms regarding system requirements. These coefficients allow the same cost function to be used to obtain different partitions, for example, one where the major requirement is reduced system area, and others where the time is considered as being the main requirement.

Another parameter studied in the cost function is the processing time. For this reason, the scheduling process is closely related to partitioning. The majority of researchers in this area consider it as being part of the process. The scheduling process deals with the order of execution of the different processes. The logical order is that which is indicated by the process graph, but in some cases, this is not sufficient as some processes may occur concurrently or other processes must wait until others finish, as they require data or commands from these.

In computer science, a scheduling algorithm is the method used to give threads, processes and data flows access to the systems resources (e.g. processor time, communications bandwidth). Hardware/software co-designs also follow this method when referring to the bus access, memory access and also the process to be computed at each moment. Within this area the possibility of several processes working at exactly the same time should be considered, as is the case with hardware modules, due to the concurrence provided by hardware based designs.

6.2.1 State-of-art

Hardware/software co-design is a technique which started during the mid nineties; it was then when the first cost function approaches appeared. Micheli and Gupta in [30], [60] proposed a formula which can be sumarized as:

$$Cost = \sum_n A_n X_n + \sum_n B_n T_n \tag{6.2}$$

In this formula, Gupta *et al* have only considered the processing time and area occupied as terms which influence the system. With respect to the time, each node is computed both on hardware and software where these values are then used in the formula. As regards the area, only the area occupied by the hardware and software nodes are considered, i.e., the occupancy of the hardware and software modules. But for both terms the communications should be considered, as the use of a hardware/software co-design requires the buses and drivers which employ both time and area to exchange data. In this approach, each module is multiplied by a different coefficient according to the designer's criteria.

P. Eles *et al* have proposed a method which assigns a weight to each node and edge [38]. The node weight considers the computational load of the node (term $M^{CL} \times K_i^{CL}$ in equation 6.3, the uniformity of its operation (i.e. replay code or hardware reuse, term $M^U \times K_i^U$ in said equation), its potential parallelism (i.e. concurrency that can be performed in hardware, therefore, reducing the computational time of the process)($M^P \times K_i^P$) and the suitability of performing such processes using software ($M^{SO} \times K_i^{SG}$).

$$W2_i^N = M^{CL} \times K_i^{CL} + M^U \times K_i^U + M^P \times K_i^P + M^{SO} \times K_i^{SG} \tag{6.3}$$

Also, edges among nodes are weighted differently to consider the communication intensity among them, i.e. the amount of data transferred, and therefore, the time expected for communication and the synchronization level among processes expressed in the number of mutual interactions among them.

The function they have proposed considers all these terms:

$$C(Hw,Sw) = Q1 \times \sum_{(ij)\epsilon cut} W1^{E}_{ij} + Q2 \times \frac{\sum_{i\epsilon Hw} \frac{\sum_{\exists(ij)} W2^{E}_{ij}}{W1^{N}_{i}}}{N_H} \qquad (6.4)$$

$$-Q3 \times (\frac{\sum_{i\epsilon Hw} W2^{N}_{i}}{N_H} - \frac{\sum_{i\epsilon Sw} W2^{N}_{i}}{N_H})$$

In this formula, three of the terms can be highlighted, these are:

- The first term measures the level of communication between the hardware and software. Reducing this term, will not only reduce the communications cost but will also favour parallelism in hardware.

- The second term makes reference to the hardware processes computed concurrently.

- The last term sends/pushes processes with high node weights into the hardware partition and those with low node weights into the software partition. As may be observed, this particular proposal is focused on time reduction by emphasizing the hardware concurrence and the hardware/software communications time.

Several other proposals have been made following these ideas, and always consider two of the constraints: time and area. However, Resano *et al* [127] have proposed a method which introduces a new term into the cost function: power consumption, this results in an interesting approximation especially when concerned with battery-powered devices.

$$F = c_a * \sum_{i=0}^{n} Area_i + c_t * \sum_{i=0}^{n} Time_i + c_e * \sum_{i=0}^{n} Energy_i \qquad (6.5)$$

Resano *et al* have proposed a separate method that considers the overheads of communications in hardware/software co-design [126]. The time computation in the cost function is based on both partitioning and scheduling. To compute this term, they assign a weight to each graph node. Initially, this weight is the processing time expected, whether it is implemented on hardware or software. After this, the execution order is then chosen for the software nodes, where this influences the weight of the rest of the nodes; and therefore, a new calculation of these weights associated with those nodes is performed. This computation cycle is first carried out on software nodes and then using the hardware nodes until none are left. As regards the area, Resano *et al.* have made a further consideration which has not been evaluated in other approaches, they have taken system drivers, controls and storage requirements as potential area sources.

With respect to the design coefficients, most proposals set these to a fixed value which depends on the complete systems requirements. However, in [115] an approach has been made which varies them according to maximum values in area or time considered. This solution helps to avoid a partition configuration where the final area or time goes beyond the viable limits.

6.2.1.1 Reconfiguration

Special attention is required for partition functions that propose solutions for reconfigurable devices. Reconfiguration is a popular hardware solution which uses the same hardware area but for different processes, this is done by changing the hardware configuration from one process to another. However, reconfiguration processes have important consequences on the time required to reconfigure and the bus occupancy needed to acquire the new bit-stream (new design for the device) from the memory where it is stored.

Initial proposals have only considered the influence of reconfiguration on the time parameter, increasing the time if a reconfiguration is to be performed. Although some other considerations must be taken into account if an architecture as shown in Fig.5.2 is used, this due to the use of the bus. For this reason Li *et al* [83] have proposed an architecture where both the software and hardware directly access the RAM memories. In this new architecture, the collision time and bus usage when the reconfiguration is carried out, do not interfere with the performance of the software, but only if the software does not require access to the RAM during this time.

Barnejee *et al* have presented a more in-depth study on the hardware/software problem associated with these devices where only part of the hardware is reconfigurable [11]. Their proposal considers several constraints which affect the reconfiguration:

- Every task considered in the hardware requires reconfiguration; when no reconfiguration is needed the task will be performed using software
- Reconfiguration resource constraints due to device size.
- No two reconfigurations can be carried out at the same time.
- The reconfiguration time should be considered before the time required for node processing.

Finally, a significant proposal has been made by Purnaprajna *et al* [118], where the authors have introduced, as part of the reconfiguration partition problem, the use of different cost functions in terms of the systems requirements.

6.2.1.2 Multiprocessors

In all of the above mentioned approaches, the general schemes have been based on the use of a single software processor versus several hardware parts. However, several proposals exist which have been designed for architectures making use of several processors within the same chip. These solutions have become quite interesting for multi-task devices which require several tasks to be performed at the same time and where most of these are sequential, this is the case for devices such as PDAs or other multithread systems. When partitioning these systems, the use of several processors must be considered; therefore, not only one process is performed by the software at the same time. Considering this, Lee *et al* have proposed a cost function to resolve the partitioning for these type of systems, considering as one of its main constraints a limited number of software processors, and therefore the number of tasks to be performed simultaneously by the software [81]. Pomante have proposed an efficient heuristic partitioning method of a multiprocessor system where the cost functions have considered the number of tasks to be performed, the affinity of these, and the communication between all the processors of the system [117].

6.2.2 Finding the minimum of the partition function

When faced with the partition problem, the equations provided are relatively straightforward although their solution is not. Regardless of the cost function considered, its solution is the configuration which minimizes the partition function found. However, depending on the graph length, millions of combinations can be considered, so, how can this minimum be found? In terms of computational complexity, finding this minimum is a NP-complete problem, and no deterministic algorithms are available to find an optimal solution within a polynomial time. Several algorithms have been proposed to solve the NP complete problems:

Approximation: Instead of searching for an optimal solution, aim for an "almost" optimal one.

Randomization: Use randomness to obtain a faster average running time, where this allows the algorithm to fail with some small probability.

Restriction: By restricting the structure of the input (e.g., to planar graphs), faster algorithms are usually possible.

Parametrization: Often fast algorithms are produced if certain input parameters are fixed.

Heuristic: An algorithm that works ”reasonably well” in many cases, however, there is no proof that it is always fast and produces permanent satisfactory results.

Metaheuristic approaches are often used. In the case of the partition problem, all the techniques which have been used to find the optimal solution have been based on heuristic solutions, as other alternatives are not feasible for the problem faced. These algorithms are:

Simulated annealing which can be interpreted as a controlled random walk in the space of feasible solutions.

Tabu search: the first meta-heuristic method which is based on a local brute force search but several intelligent tools are introduced to override the already examined solutions or move to another solution space area.

Genetic algorithms , these are based on the natural phenomena of species evolution, in which the genetic modification is assumed to lead to better specimens, in our case, towards solutions.

6.2.2.1 Simulated annealing

This method receives its name as it works in a similar manner to the temperature variation during an annealing process [134], [59] and [154]. It relies on the modification of an external parameter, called temperature, which varies in value according to the minimum expected:

1. The temperature parameter is fixed to its initial value.

2. Initially, a set is examined. This first set is obtained randomly. This is referred to as the current set.

3. The actual set’s neighbour is generated randomly.

4. The values of both partition functions are computed as well as their difference, $\triangle C$:

$$dcost = cost(neighbourset) - cost(actualset) \tag{6.6}$$

5. If this difference is below 0, the actual set is updated to the neighbouring set, and the process continues starting with step 7.

6. If not:

    (a) A random choice is made for a q value between 0 and 1.

    (b) If this value is bigger than the system energy $e^{(\triangle C/T)}$, then this process returns to point 3. If not, the actual set is updated to the neighbouring set we are working with.

7. The temperature value is decreased. This value can be updated depending on different algorithms.

8. If the temperature value obtained is still higher than the cooling temperature, a new set from the neighbourhood of the examined set is used in step 4.

The temperature variation determines the feasibility of the algorithm: if the variation is very fast, only a few sets will be examined, and therefore, the system will be relatively fast. However, there is a high probability that the minimum cost configuration will not be considered, and thus the result obtained is not optimum. As previously explained, the algorithm works quite well when searching for local minimums, however, there is an important absence in the algorithm: if the minimum is not located in the initial neighbourhood set, the algorithm is not likely to find it, especially when there is a maximum between the two minimums. To solve this problem, Wiangtong *et al* have introduced a further modification into the algorithm of the temperature value [154]. If the value obtained using the partition function with the set studied is higher or significantly higher than the minimum already stored, the temperature value increases and thus a wider solution area is explored. Using this solution, it is easier to move around different minimums. The main disadvantage is the significant increase in time required to find a solution.

6.2.2.2 Tabu search

This search is similar to the simulated annealing technique; the tabu search is capable of finding the function minimum when a termination condition is satisfied. The main difference between this and the previously explained algorithm is that in the tabu search algorithm, Glover (its creator) has introduced memories to obtain a more intelligent algorithm than the previous one. These memories can be classified as short term or long term memories [55], [9], [110], [54] and [18].

The short term memory is characterized by a list of tabu movements. This list is formed by several movements which are forbidden to be taken in the iteration being dealt with. Highlighting among these forbidden movements are those previously computed, thus avoiding examining sets which have already been investigated.

In order to make this list finite, the number of elements in this list is reduced in each iteration. When a movement becomes part of the list, a restriction value is applied to this set. This value usually decreases as more sets are examined, until an iteration with a null value is found. Therefore this set leaves the list and the tabu state.

To leave the list, the aspiration criteria is introduced to allow a solution to be re-examined, even though it is considered as tabu. The aspiration criteria is an algorithm which revokes tabus, usually because the result obtained with that tabu solution is better than the current best-known solution.

As opposed to these lists, long term memories provide a suitable solution to perform a more in-depth examination of several solution areas, named promising areas, where it is expected to find the optimal solution (intensification) or furthermore, to avoid continuous searching within the same areas, by moving to unexplored areas of the solution space (diversification).

Fig. 6.3 shows the dataflow of a generic tabu search algorithm. In this example, no memories other than the tabu list have been considered. It may be observed that initially it works similarly to the annealing technique, but once the values of the functions of the initial set are computed, it verifies if the movement with better results is in the tabu list; if it is not, it is considered as the best solution. If the movement is in the tabu list, the aspiration criteria is examined to check if the solution examined satisfies it and therefore, considers it again as the best solution.

6.2.2.3 Genetic algorithm

A genetic algorithm (GA) is a search technique used in computer science to find exact or approximate solutions for optimization and search problems [67], [153] and [154]. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (chromosomes or the genotype of the genome) of candidate solutions (individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions [3]. Traditionally, solutions are represented in binary as

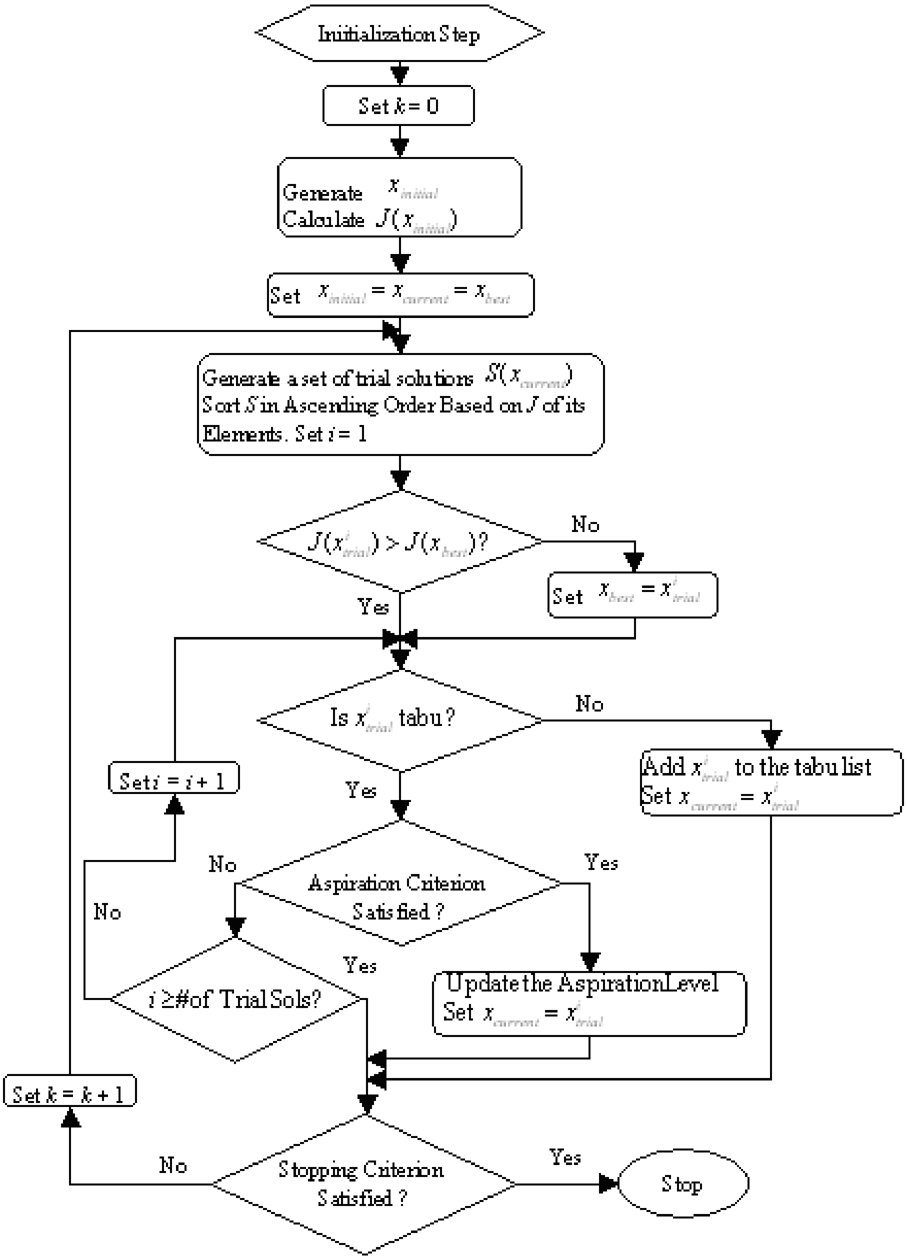

Initialization Step
Set $k = 0$
Generate $x_{initial}$
Calculate $J(x_{initial})$
Set $x_{initial} = x_{current} = x_{best}$
Generate a set of trial solutions $S(x_{current})$
Sort S in Ascending Order Based on J of its Elements. Set $i = 1$
$J(x^i_{trial}) > J(x_{best})$?
No
Set $x_{best} = x^i_{trial}$
Yes
Is x^i_{trial} tabu?
No
Add x^i_{trial} to the tabu list
Set $x_{current} = x^i_{trial}$
Set $i = i + 1$
Yes
No
Aspiration Criterion Satisfied ?
Yes
No
Yes
$i \geq$#of Trial Sols?
Update the Aspiration Level
Set $x_{current} = x^i_{trial}$
Set $k = k + 1$
No
Stopping Criterion Satisfied ?
Yes
Stop

Figure 6.3: Tabu search algorithm flow chart [66]

strings of 0s and 1s, but other encodings are also possible. Evolution generally starts from a population of randomly generated individuals and occurs in generations. In each generation, the fitness of every individual in the population is evaluated; multiple individuals are stochastically selected from the current population based on their proximity to the best solution (fitness), and modified by recombination to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm terminates due to a maximum number of generations, a satisfactory solution may or may not have been reached. Fig. 6.4 shows the flowchart of a genetic algorithm. The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. In some problems, it is hard, or even impossible, to define the fitness expression; for such cases, interactive genetic algorithms are used.

6.3 Bottlenecks

To finish this chapter we would like to highlight the problems associated with the co-design architecture indicating how these drawbacks are dealt with. Considering the different proposals described throughout this chapter, and the architecture shown in Fig. 5.2 ,where details were given in chapter 5, both hardware and software share the same access to peripherals using the same two buses: the data bus and the control bus. The occupancy of these buses' is one of the major problems associated with this type of system, as potential collisions may occur here, especially when access to the same peripherals is desired. This problem is predominantly seen at the input/output of the peripherals and memories.

Sharing memories is the major bottleneck these systems have. As it can be observed, both hardware and software may access these memories at any moment, this is particularly true for the RAM memory which contains temporary data during the execution process. These memories are based on a sequential access, thus when they are occupied by one processor, other processors must wait until the first processer has finished, especially when the process is data dependant, i.e. where one processor requires the data provided by the other. In this case, the scheduling process should consider this dependency and establish a waiting time for the latter processes until the first finishes. If the case arises where both processes desire memory access only to read

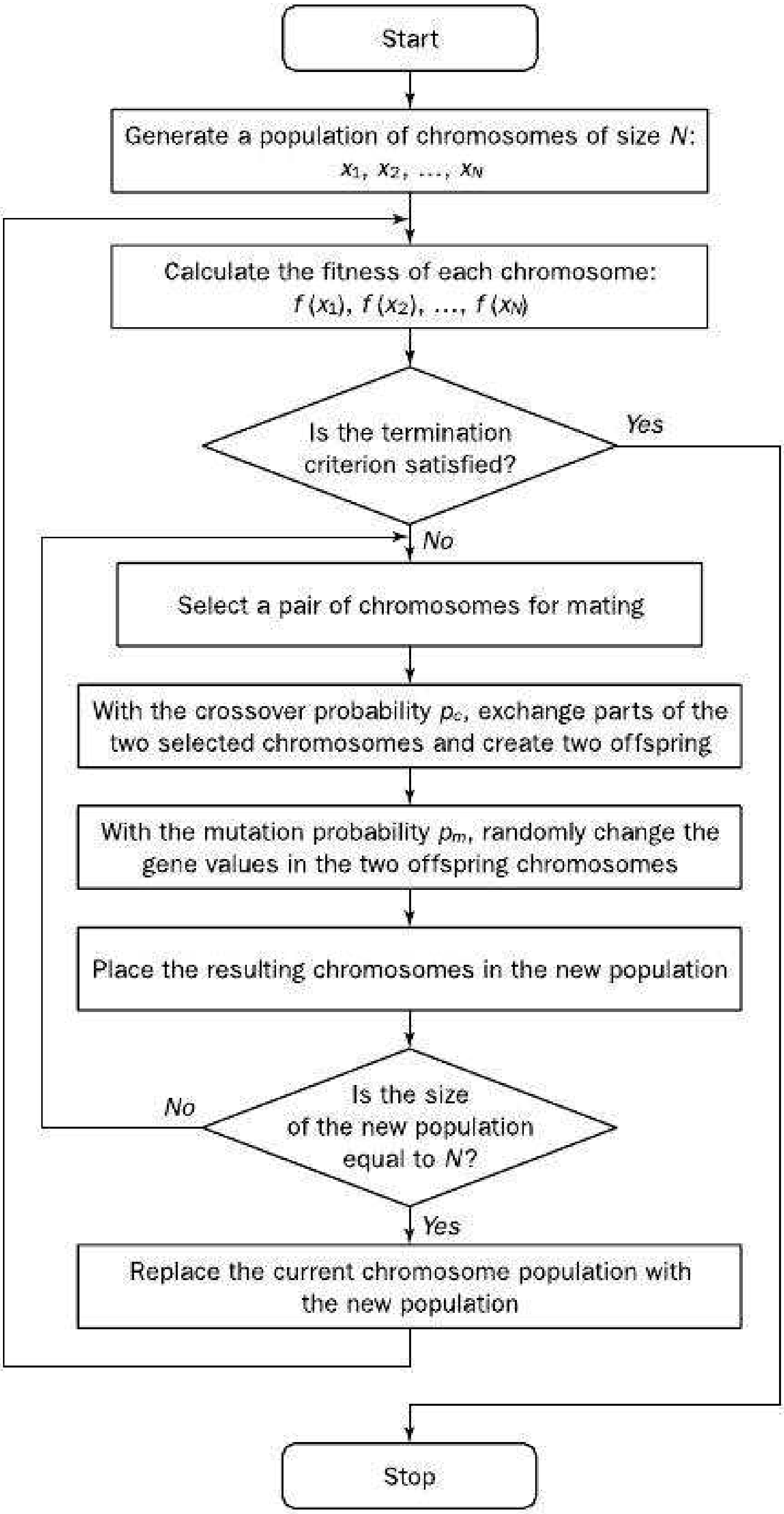

Start
Generate a population of chromosomes of size N: $x_1, x_2, \ldots, x_N$
Calculate the fitness of each chromosome: $f(x_1), f(x_2), \ldots, f(x_N)$
Is the termination criterion satisfied?
Yes
No
Select a pair of chromosomes for mating
With the crossover probability p_c, exchange parts of the two selected chromosomes and create two offspring
With the mutation probability p_m, randomly change the gene values in the two offspring chromosomes
Place the resulting chromosomes in the new population
Is the size of the new population equal to N?
No
Yes
Replace the current chromosome population with the new population
Stop

Figure 6.4: Genetic algorithm flow

it or access different sections of it, the access should be sequential and therefore, bus arbiters are commonly included to control this access.

Another important problem that arises is related to the communication between the hardware and software platforms. Data transferred between them should follow the same arithmetic. However, since software generally works using floating point operations and hardware solutions are usually based on fixed point arithmetic, then the information should be transformed prior to the exchange.

Designing embedded systems following approaches based on hardware/software co-design has been pointed out as a difficult task due to the many variables which should be considered. Apart from these considerations, the designer is faced with many problems regarding the potential architectures and their combination. The mixture of different architectures and solutions has demonstrated many advantages when compared to single architectures but the number of problems it entails also increases. We have presented in this chapter the state-of-art in partitioning and scheduling. Both problems characterize the hardware/software co-design. We have presented how designers have faced the problem of selecting which parts of the process should be performed using hardware and which using software. Following these approximations, we will present later on this Thesis our proposal to perform this process considering the particular characteristics of a biometric ID token.

Chapter 7

ID tokens

Throughout this Thesis it has been stated that ID tokens can be used as devices for identification purposes, however no further explanations have been provided thus far. As mentioned in the objectives of this Thesis, a new model proposal has been built and details will be provided in the following chapters 9 and 10. But before providing details on this new proposal, it is necessary to explain the workings and functions of these devices, their environment and their requirements. In this chapter, all these areas will be discussed, including the evolution of these devices and the novelties that have been introduced since they were first created. Finally, the functionalities required by these devices will be presented, and the requirements that should be satisfied to achieve the required amount of security and reliability.

7.1 ID token definition

As mentioned in previous chapters, ID tokens are embedded devices used for identification purposes. However, up until now no clear definition of what an ID token is has been provided. An ID token is a device which contains user information which can be used for identification purposes. Many types of ID tokens can be found which are based on different technologies and include a wide variety of security mechanisms: from sophisticated tamper-proof biometric systems, to simple business cards.

Initially these tokens were used to provide information on the user, such as name or profession. However, with the passing of time, ID token technology has advanced and so to has the possibility of token forgeries. ID token functionality and ID technology have also progressed accordingly: from visually presenting the data to storing it electronically, as well as several protection mechanisms incorporated to inhibit potential attacks and forgeries; these devices are now able to perform identification processes by themselves. Thanks to advances in technology and the increase in storage capacity, ID tokens have become a major part of distributed architectures, where data stored on these devices is easily transported by the user, and presented for identification purposes, this reduces the requirement for centralized databases [140]. The current architecture of these systems is shown in Fig 7.1:

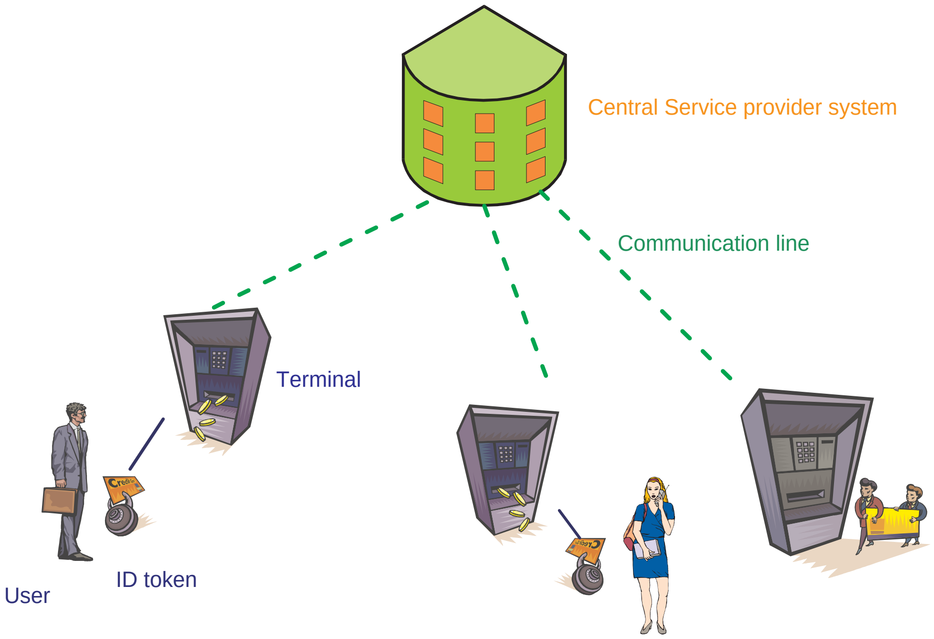


Figure 7.1: Distributed system based on ID tokens

In these systems, each time the user requires access to a certain facility, the ID token should be introduced into the system terminal. The terminal, using the information stored in the token, will perform the identification process and so, allow the user access if the identification is positive. If not, after a certain number of consecutive failed attempts the token will be rejected by the system, requiring the user to contact the system administration services. Generally, these systems are more secure than those based on centralized architectures where a central database stores all the data, and thus, provides a weak point with respect to the security as a lot of information is compromised during a successful attack. A successful attack on a distributed system only compromises the information stored on a single token, if the user is aware of this attack the system administrator can take the corresponding countermeasures.

Distributed systems are composed of the following elements:

The Service Provider System: although the information is not stored here, these systems are used to control the access of users and provide all the relevant services to the clients. As regards user control, the central system can record operations performed by the user, i.e. user history. This central system can be located anywhere and communicates with each of the terminals. This communication should be secure and in some cases, it is built using specialized communications channels [32].

Terminals: each of the access points to the system are known as Terminals. These devices may be located anywhere they are required, i.e. at the entrance to big facilities or at any Point of Service. These terminals may have different shapes and function differently, depending on the ID token technology used. The main feature of this equipment is that the terminal is capable of reading the information stored in the token and transmitting it to the central system via the communications line channel. This communication line channel may or may not be a specialized link, but the information transferred should always be encrypted or at least protected against potential modifications. The terminal, additionally, may perform further functions, such as acquire biometric data and also process this data for later comparisons with the stored templates. Other commonly used operations may also be implemented such as an ATM to provide money or even to open a gate/door.

Tokens: the main function of the token is to store the users personal data, including information such as bank account details, name, nationality and biometric data. The token should provide the stored information to the terminal, and always ensure secure information transfer. Additionally, some other functions can be included in the token such as external identification, using a photo or signature which can be compared visually with the user. These tokens can provide extra security countermeasures if they contain cryptographic keys and/or tamper-proof mechanisms. The combination of tokens and cryptography increases the systems security, therefore avoiding the drawbacks of using centralized architecture systems.

For the enrolment process, it is important to consider the participation of a third party who can certify the users' identity as being part of the service provider. This third party is known as the certification authority and in most cases, is a government agent such as a police officer or administration personnel. The certification of the user identity is only considered during the enrolment process, when the user provides his identification, and the service provider accepts it as being true.

In this chapter we will focus on tokens, while keeping in mind the use of the terminal as the interface between the service provider and the token.

7.2 Classification of ID tokens

ID tokens can be classified according to different criteria, such as the technology used, or the identification mechanism they rely on.

Based on the technology used, several types of devices can be found, these are[32], [15]:

Simple non-electronic devices: These tokens are made of paper or plastic, such as business cards, photographs, recommendation letters, etc. These tokens are only used in cases where the user identity is not required to provide any further services and is limited to when the user introduces him/her-self to another person. These tokens are easily forgeable and as a result the data provided is basically for information purposes.

Complex non-electronic devices: considering the security drawbacks, early token technology, before hardware integration was possible, was based on ID tokens with no electronic mechanisms but did provide several security countermeasures to complicate potential forgeries. These devices provide complicated patterns, holograms or magnetic stripes, which make counterfeiting a difficult task. Among these, it is important to highlight credit cards or older versions of ID documents.

Electronic devices: When transistor integration became sufficient enough to develop small electronic devices with acceptable storage capacity, the first electronic ID tokens appeared. These devices are not only able to store identification information but are also protected using cryptographic techniques. Within this field it is important to point out the use of smartcards which are currently being used as ID tokens that may also be combined with biometric data. Other ID tokens of this kind are based on portable memory devices, such as pen-drives, which only provide access to the data they store if the user is first identified by a PIN-code or a biometric trait. These devices can also be divided into different categories depending on several criteria, these are:

- Considering the connection to the terminal:

- Contact: these devices require direct contact with the terminal to send and receive information. As a result of this connection, these devices receive the required power from the terminal to function.
- Contactless: These devices usually communicate with the terminal by means of wireless protocols. For power purposes, the majority of these devices use a coil to generate the energy required. Common examples of these devices are proximity smartcards, used for access to several big facilities, and chips for animal recognition such as cow tags or pet chips.

- Considering the data protection mechanisms included:
  - Low protection: such as the case of animal identification, the data these tokens store can be read by a terminal without any kind of additional security.
  - Medium protection: In some cases, the data can only be read from authorized terminals and therefore provide limited access. This is the case for many proximity cards used to access facilities such as garages and public transportation services.
  - High protection: to access the data stored, the terminal and the users must provide more information which is only known by the user, i.e. cryptographic keys or a user pin. These tokens can be divided into several groups which consider additional security measures used for data protection:
  - Those which use cryptographic techniques:
    - Hard tokens: these are hardware devices which contain a protected cryptographic key. For key release, the user or the terminal should provide a password or a biometric feature to activate this key. These devices are not able to export the authentication key, thus, the key is always stored and managed inside the token, this provides additional security.
    - Soft tokens: The token key is encrypted under a key derived from activation data (pin or biometrics). This is the case, for example, for cryptographic keys provided from a biometric feature vector or a password that is only known by the user.
    - One-time password device token: these tokens are able to generate a password each time it is used. For this purpose, a ciphering block

    is usually included in the token to combine a private key stored in the token device and the one generated at that instance of use.

    - Password token: The password is associated with the token, the user must memorize this password, this avoids several impersonation problems.

- According to where the comparison process is carried out: These devices can perform the identification themselves if they contain processing capabilities. As a result of this possibility, the tokens can be further divided into two main areas of operation:

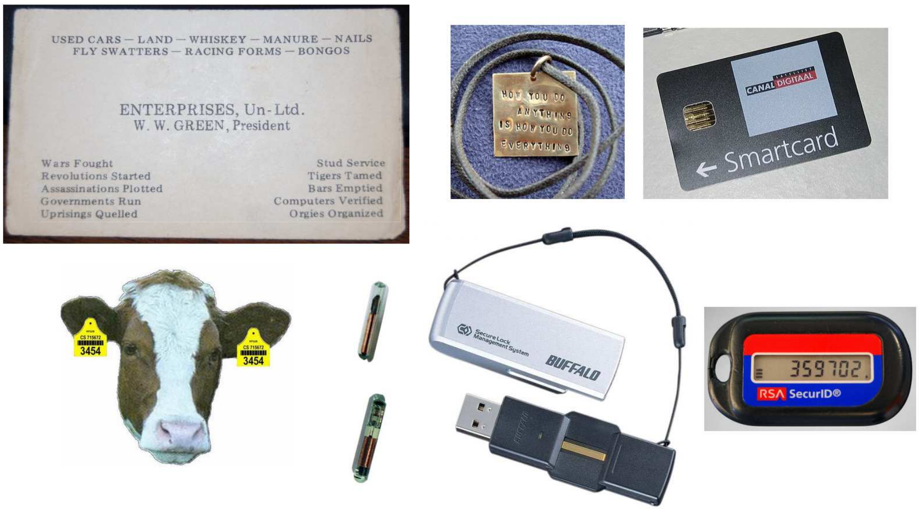
    - Comparison-on tokens: Here the token performs the verification within the token itself, thus the template is stored in the device where no transmission of sensitive data to the terminal is required. However, the terminal is still in charge of acquiring the biometric data and processing it; once the feature vector is obtained it is sent to the token for the comparative analysis.

    - Store-on token: In this case, the terminal performs the comparison. In these systems, the terminal asks the user to provide the biometric raw data, processes it and when the feature vector is obtained, the terminal asks the token for the template which is then transmitted to the terminal. Once the template is available, the device performs the comparison. This architecture is widely used as the token does not require any computation capability but only storage capacity of the biometric template. However, these systems have been proved to be more vulnerable to attacks than the previous architectures, this is due to the exchange of biometric information.

Any combination of the above alternatives: in order to protect the information stored and the users' identity, a combination of the above mentioned mechanisms can also be found. This is the case for some passports or national identification cards.

7.3 ID token evolution

The first ID tokens date back to the 14th century in China and France, where both governments issued safe-conducts for their population when travelling. Safe-conducts



USED CARS — LAND — WHISKEY — MANURE — NAILS
FLY SWATTERS — RACING FORMS — BONGOS

ENTERPRISES, Un-Ltd.
W. W. GREEN, President

Wars Fought
Revolutions Started
Assassinations Plotted
Governments Run
Uprisings Quelled

Stud Service
Tigers Tamed
Bars Emptied
Computers Verified
Orgies Organized

NOT YOU DO ANYTHING IS HOW YOU DO EVERYTHING
CANAL DIGITAAL
Smartcard
3454
3454
Secure Lock Management System
BUFFALO
359702
RSA SecurID®

Figure 7.2: ID tokens examples

can be considered as the first ID tokens as they provided the users information which was certified by a trustworthy third party; in these cases, the relevant government administration [103], [85].

Other paper ID tokens can be found throughout history, where business cards are currently one of the most common examples [137]. However, although these cards provide the users name and profession, they are not certified by any external body, thus there is no credibility to the information they contain. Due to the large number of cards being used in different systems and due to the start of forgeries, increased security has become an important issue. Also, these types of ID tokens were required to process information electronically without the use of operators reading the documents. For this reason, plastic cards have been substituted by magnetic strip cards, which provide information stored in a magnetic band. These cards are widely used nowadays as banking cards. The first card used for this purpose was the ”Diner’s Club” in the USA. This card was used by a group of businessmen for shopping transactions and to pay restaurant bills and were used as a means of direct contact with the users bank. The widespread use of these cards prompted the emergence of the first credit card companies, i.e. MasterCard and Visa, which, after some time, incorporated more secure means such as patented colours in their prints or photographs on the cards. The most important add-on was the use of a magnetic band. This band situated at the back of the card stores the users information such as bank account details, the users name, etc. However, the usage of this band provides several drawbacks such as the limitation of information storage space, the lack of protective countermeasures for the information stored and the loss of the data stored in the presence of magnetic fields.

As a way of combating all these defects the development of smartcards appeared. In this type of card the most prominent add-on is a chip formed by memories and a microprocessor. Thanks to this new technology, these cards are capable of storing up to 16 KB of data and process several security processes such as information ciphering and deciphering, and even the possibility of performing biometric comparisons. The use of these cards has been widely extended in everyday life and are commonly used for everyday tasks such as access to public transport (e.g. Poland and London), opening garage doors, EMV credit cards and animal identification.

Unfortunately some processes cannot be performed using these cards due to the reduced processing capabilities. It is true that they can store biometric data and compare these with a template, but when attempting to perform further identification processes, these tokens are unable to do so. For these reasons, new approaches have been undertaken in the past years to increase the computational power of these devices, this is especially the case for dedicated ID tokens.

7.4 Functional requirements

Up until now, the basic functions of the ID token has been presented, where this has included a historical review of their evolution. In this section, the functional requirements of the ID tokens will be discussed, the features they should pertain and the requirements to be covered to obtain a reliable solution [124], [140] and [109];

Portability An obvious requirement is related to the size of the device. An ID token should be portable, allowing the user to carry it comfortably. It's size should be as small as possible and should also be lightweight. Commonly used ID tokens exist with different shapes and sizes, from the cards we use every day to tiny chips situated under a dog's skin.

User friendly Since ID tokens may be used many times on a daily bases to gain access to a particular system, the interaction with the user and the terminal should be as user friendly as possible to prevent any uncomfortable feeling that may result and which may lead to technology rejection.

Real-time solution User access to a system should be as fast as possible. In many cases, it is not permissible to have users queuing up, awaiting identification. This scenario is commonly seen at airports and border controls, where time-consuming identification procedures retain users excessively at control points.

Power consumption The power consumption of these devices should be minimal, as the addition of batteries increases the size and weight of the device making it more inconvenient for the user to carry.

Storage requirement The token should be able to store all user data that is required, and also if used, cryptographic keys. For this reason, an EEPROM memory is used as it is important that the information is not erased when the devices is un-powered. The size of this memory depends on several factors: if the device is to contain a cryptographic key, the size of the biometric template, etc. When using Iris Biometrics, the storage capacity required for the complete set of biometric information is less than 2KB, which is much lower than in other biometric modalities, such as fingerprint correlation algorithms, where storing an image as a template is required.

Processing capabilities The processes designed to be executed by the token determine its processing capabilities. When using cryptography the token should be able to cipher and decipher the information exchanged. When using Biometrics as a comparison-on-token device, the comparison process should be performed within it. This generally signifies very different requirements, i.e. the use of simple algorithms such as the Hamming Distance, or complex processes such as Gaussian models or neural networks [137].

7.5 Security requirements

As well as the functional requirements, the ID tokens should satisfy several security requirements as the information they store is highly sensitive. These requirements are related to three main topics [124], [16]:

- Communications Security
- Physical Security.
- Electronic hardening.

7.5.1 Communications security

When the system requires a high level of security, due to potential forgery risks of the information stored, or due to the sensitivity of the services it provides, the system should follow the PKI (Public Key Infrastructure) scheme. Apart from PKI there is also symmetric cryptography, where the communication confidentiality is based on just one cryptographic key which is known by both the user and the users service provider (symmetric key). PKI secrecy is based on two keys per user and/or token:

- The public key, as its name points out, is public and can be known by everybody.
- The private key is only known by the user, where this user is in charge of storing and maintaining its secrecy.

Both keys are mathematically dependant, although it is important to point out that just one of these is known and the other one is widely known, and due to their mathematical relation, the private one is not deducible. This is commonly referred to as asymmetric cryptography.

Tokens cipher the information they send using the private key, which can only be deciphered using the corresponding public key. Therefore, both the service provider of the transmitted information and the user's public key may receive this information at the same time, confirming the transmitter identity, i.e. the user. Also the vice-versa situation arises where the terminal can provide information to the token, encrypting it by means of the public key which can only be deciphered by the token with its private key.

With the key provided once the recognition is performed, the token encrypts all further communications. The algorithms used for encryption vary from one token to another. Among these algorithms, the following two are currently the most recommended for use:

- RSA: Asymmetric. The length of the key is quite long and requires both a public key and a private key; for this reason, this algorithm is widely used in several National ID's such is the case for the Spanish system.
- AES: Symmetric. This algorithm is been used as a standard by the American government to encrypt their national identification cards or passports. There have been several attempts to decipher codes encrypted by this algorithm without any success.

7.5.2 Physical requirements

Even when protecting the communications link with the terminal, the token is still a physical device that can be subject to attacks, such as reverse engineering [76][107]. The main intention of these attacks is to access physical parts of a token to read information and whatever processes these perform. Physical countermeasures should be considered when creating a token to protect it from such attacks.

- Make physical intrusion obvious, the external casing needs to be breached to access the internal components: any potential attack of this kind ideally should leave evidence of it, such as scratches or pry marks.

- Make internal tampering externally visible: several tokens provide different means of showing the presence of internal intrusion such as changes in the colour of a tamper-proof window, zeroing the EEPROMs or tripping a ”dead-man’s switch”.

- Encapsulate internal components in epoxy. If the intruder manages to access the token, the internal components should be protected using an additional encapsulation. Epoxy resin increases the difficulty of access to these components, where to remove the epoxy heat is required that would most likely damage the encapsulated electronic components.

- Use glue with a high melting point: glue with a high melting points is similar to using epoxy in the encapsulation.

- Obscure part numbers: if the chips are recognizable, possible reconstruction of the board can be carried out.

- Restrict access to the EEPROMs: EEPROMs store non-erasable data, and thus, personal information or cryptographic keys. Therefore, the access to these memories should be as limited as possible to avoid potential intrusions where this data can be read and later replicated.

- Remove or deactivate future expansion points: expansion points are commonly used in hardware for later modifications, debugging or expansion. These may be access points to data and processes stored within the system, and therefore, should be removed.

- Reduce electro-magnetic frequency (EMF) emissions: the information transmitted between the token and the terminal can be deduced from the time spacing between signals, and the electronic-magnetic pulses emitted.

- Use good board layout and system design: Distributing the connections between the components on different layers increases the difficulty of possible eavesdropping of the processes that are taking place within the token, and therefore avoids replication of the processes.

7.5.3 Electronic hardening

The electronic countermeasures that a token should consider are related to potential electronic attacks such as firmware modification or other potential changes in the electronic parts :

- Protection of flashable firmware: potential changes of the firmware may lead to the key stored in the system being released. If the firmware is changed, the identification process can be avoided by introducing a firmware which always leads to a positive identification. For this reason:
  - Firmware should be encrypted and decrypted by the device before updating it. Additionally before accepting this firmware, both the token and terminal should recognize the rejection of firmware from other devices or intruders.
  - The updated firmware needs to be signed digitally.
  - The firmware should be compiled-optimized before being released, and all debugging and symbol information should be removed.
- Integrity of communications with the terminal: The communication between the terminal and the token should be encrypted before being sent. The data should also be signed and time-stamped to prevent an attacker from recording a transaction and playing it back later. To accomplish this, a shared secret is required. A shared secret can be transmitted clearly, however this may be intercepted by tapping the communications channel.
- Protection from external injections of spurious data: at the I/O points of the system spurious data can be introduced to compromise the system or impede its use. Different attacks related to I/O should be considered:
  - Buffer overflows: when receiving data, this data is usually stored in a buffer. If the amount of data received is superior to the buffer capacity, information collapse or overwriting is possible.

- Use of undocumented command sets and functionality: The communication between the terminal and the token is done using a number of burst bits. These bits should be interpreted by the token as commands. When the bits do not correspond to any specific command, the token should be able to realize this and not perform any actions interpreting it as a possible attack, denying access to the information and the services provided.
- Inappropriately structured data elements.
- Improper failure states: when a token goes into a possible error state, a method should exist to recover from such an error. If not, potential denial of use may occur or the functionality of the device may be altered.

Throughout this chapter the different areas corresponding to the design and development of ID tokens have been discussed in detail. The embedded nature of these devices conforms to its portability and single task operation of storing identification data and in some cases performing this identification. However, and as presented, other considerations should also be taken into account; these are the operational environment, security of the information contained within the device and the security of the device itself.

Part III

Contributions

Chapter 8

Designing ID systems based on ID tokens

Throughout the previous chapters different aspects of Biometrics and embedded systems have been studied. The work developed and presented in this Thesis is based on the combination of these two different areas leading to two main contributions: The first is related to an optimized ID token design where all the different aspects of these devices have been considered. The second has been the implementation of a search engine whereby efficient hardware/software architectures have demonstrated exceptional results as regards accelerating the comparison process and therefore, making it applicable for larger databases.

In this chapter, the proposal for the design of a recognition system based on ID tokens is presented, along with their analysis and the development of this device as an embedded system. Many researchers have already presented proposals based on different combinations of hardware for biometric purposes using such devices [89], [48] and [37]. However, these approaches have only considered the acceleration of high computational processes due to hardware concurrence or efficient use of resources. The motivation behind the work presented here has been to go beyond previous approaches and consider not only increased speed but also other advantages such as power consumption and security features.

8.1 Motivation

ID tokens have been widely used and developed during the last few years. The addition of further security measures for these devices has also increased, preventing potential forgeries. At the same time, ID tokens have become more complicated, not only because of security countermeasures but also due to significant changes in the identification process. Biometrics has become an important technology for recognition purposes because of security and the possibility of using this modality even when the user does not want to be recognized (negative recognition and non-repudiation). Combining these two technologies is an important requirement for current identification applications. Several manufacturers have developed systems where Biometrics has been included in tokens where these are based on microprocessor-platforms [68], [100], [99], [147] and [102]. However, these solutions are only capable of performing the comparison process or storing the biometric data and are not capable of performing additional biometric processes due to the low processing capability.

Hardware/software co-designs can satisfy these necessities by including, within the token, dedicated co-processors which perform high computational load processes. These solutions do not only provide this particular advantage but furthermore are able to perform almost any function in less time than software-based solutions. Another important feature is the reduced area required when using dedicated hardware. However, several questions arise: What is the best choice of partition? Can hardware provide our design further features than solutions based on software? Currently, several approaches have been proposed which deal with the combination of Biometrics and hardware by developing unique co-processors for specific biometric processes. However there has been no generic proposal to design these independently of the biometric modality used or the specific algorithm. This has been the major motivation behind the following contribution: We have performed an in-depth study on both sciences where we have attempted to approach them in order to create a design methodology to develop an ID token based on a hardware/software co-design by considering all the different features surrounding these devices.

In this chapter a discussion will be presented on several key points related to the design of the complete biometric system based on ID tokens. For this purpose, this chapter is organized as follows: first a description will be provided on the different features these systems should fulfil by considering the environment within which they are used. We will describe the different parts which form the system by focusing on the use of an ID token, as this is where we have developed the majority of our work. The token communication protocol between the token itself and each terminal is described in section 8.2. The methodology proposed to design the ID token based on a hardware/software co-design is described in section 8.5, as well as the different considerations made for this proposal.

8.2 General system architecture

The general system architecture is based on a distributed system where biometric and other personal data is stored on tokens. Each user is provided with an ID token during the enrolment process. The decision to use a distributed system has been based on the security advantages they provide: having one unique central database can compromise all user data during a successful attack; by providing each user with a token with their data, the Orwellian connotations of biometric systems are minimized. The system has been formed using a central system which communicates with the different tokens and all of the different terminals, where each terminal is situated wherever required.

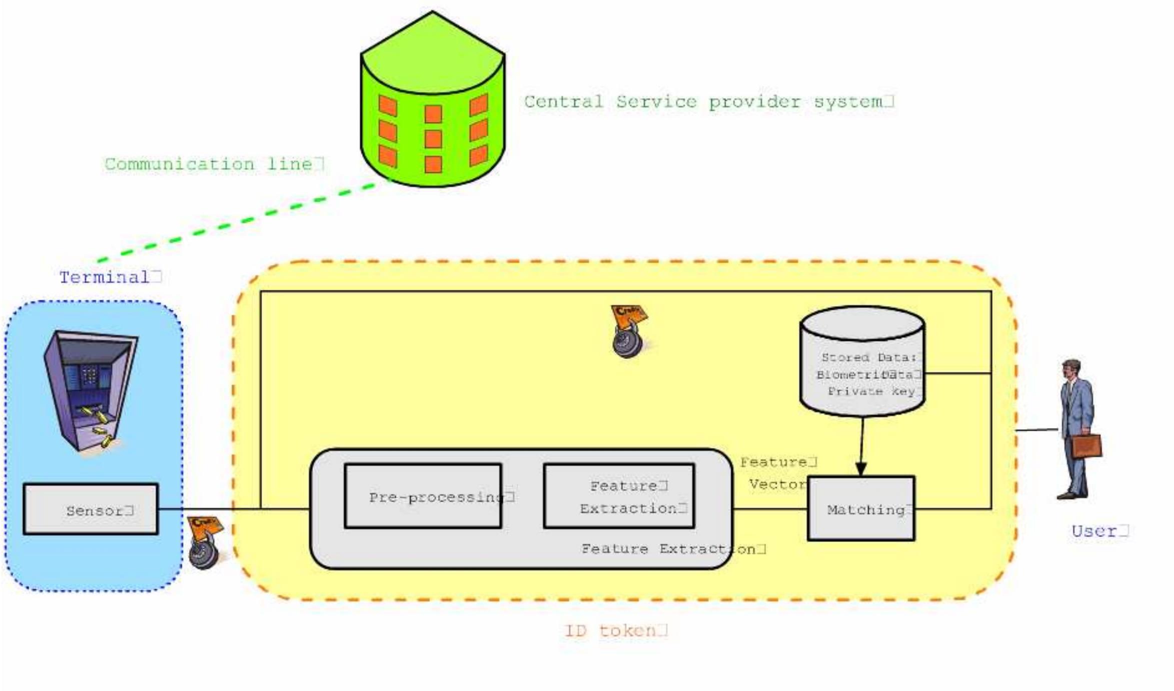


Figure 8.1: Architecture Proposed for a Verification System

It is recommended that the communications link between each terminal and the central system is encrypted using algorithms with a low probability (or null) of being decrypted. This algorithm may also be the same as that used for communication between the terminal and token. The central system is in charge of registering access and the operations performed by users and the services they request. This registration provides a backup for possible errors and impersonation access which may lead to loss and theft of user information. Although the research presented in this Thesis has been focused on the ID token design, the system terminal must also be considered, as it is directly related to ID token. The terminal in our system proposal is also susceptible to intruder attacks; therefore, the less amount of functions it performs the more secure the complete system will be.

The function of the terminal in the current proposal is limited to acquiring the biometric raw data and all the functions related to the central system communication and further services. The terminal will not perform any processing related to the recognition or storage of user data. The terminal architecture depends on the function of the complete system and the biometric modality used, and thus, the configuration used depends on these aspects. Including one or other biometric modality depends on the working environment of the terminal, as not all the modalities present satisfactory results in every environment due to their performance and the behaviour of the sensors, such as illumination or degradation.

The terminal, as we mentioned in chapter 4, is one of the possible entrance attack points regarding artefacts, synthetic samples and other attacks related to forging the biometric data. Our solution does not implement any method or techniques against this type of attack, as these are not possible to solve by means of hardware architectures. However, we recommend algorithm countermeasures for these attacks, such as adding aliveness detection algorithms or increasing the terminal functionality by including additional measurement sensors.

8.2.1 Communications between the terminal and token

A communications link should be established between the terminal and the token each time they interact. As has been mentioned previously, this link should be secure to avoid possible man-in-the middle or replay attacks. Moreover, the communications link between each device must follow specific protocols for correct and secure information transfer. This protocol will be described here.

The frames sent by the terminal to the token have the structure shown in table 8.1:

Command	Length	Data	CRC
8 bits	8 bits	...	8 bits

Table 8.1: Command Frames Structure between the terminal and the token

The length of data field is determined by the length indicated in the general data frame.

The commands implemented are pointed out in table 8.2. As can be observed from this table, distinction is made between two different types of commands: those where the most significant bit is ’0’ and those with ’1’. The first group is formed by those commands that are carried out during the identification process, where any terminal can ask for these; the second group can only be carried out during the enrolment process or during system updates. This second group should be performed in a secure environment to avoid the modification of token data.

Op Code	Description
00000000	Sending image
00000001	Start identification process
00000010	Stop process
00000011	Ask for the token status
10000000	De-block token
10000001	Number of failed attempts?
10000010	Reset token
10000100	Update biometric template
10000101	Update biometric template
10000110	Update cryptographic keys
10000111	Update biometric threshold

Table 8.2: Commands between the terminal and the token

The token communicates with the terminal according to the same frame, but instead of the command table described previously, the token provides the answers shown in table 8.4.

Op Code	Description
00000000	Provides the token state in the data field: idle, processing or blocked.
00000001	ID result in the data field: positive or negative identification.
00000010	Return the number of failed access attempts.
00000100	Return the last template computed which led to a failed attempt.
10000000	Token error due to incorrect sent data.
10000001	Pre-processing block error.
10000010	Feature extraction error
10000011	Overflow error
10000100	Unknown command

Table 8.4: Answering Frames from the token to the terminal

As in the previous case, the response of the ID token can be divided into two main groups, these are indicated by the first bit of the command: correct/normal processing or errors which are indicated by a '1' as the first bit. Information can be obtained from the token during normal processing responses, from the result of the identification process or information from this process. The errors considered are those obtained during the identification process steps, such as in the pre-processing (no iris found in the image, the quality of the image is not sufficient for identification purposes, or the values obtained are erroneous), in the feature extraction block (null iris signature) and overflow errors or unknown commands.

8.3 General architecture of an ID token

As has been mentioned in the previous chapter, several considerations should be made when designing an ID token, especially when dealing with the security. An ID token should be capable of performing identification processes while at the same time securing the data it stores as well as the data transfers. Due to the tokens environment and architecture, our proposal attempts to reduce the risk of attacks.

Up until now, the majority of biometric tokens only perform matching-on processes, i.e. the comparison process is performed in the token to avoid template transferral outside the token. With this solution, the system is increasingly secured as no template is transferred and any intrusion requires direct access to the token. However, these solutions are still insecure. An intruder, by performing a man-in-the-middle attack between the terminal and the token, can obtain a feature vector sample which may lead to a correct identification that can later be replicated. For this reason, the addition of further biometric processes has been considered necessary during the development of the ID token. Thus, we have included the feature extraction and the pre-processing block to reduce the likelihood of possible authentication attacks. The addition of the sensor in the device is possible for a limited amount of modalities such as fingerprint sensors which are based on capacitive sensors. However, in most cases, sensor incorporation is not plausible due to size limitations. For small portable devices it is unreasonable to include sensors such as a camera which is required for iris recognition or a tablet for signature recognition. The architecture proposed is as follows:

The different elements included in this architecture are:

Serial interface: the token should be capable of communicating with the outside world, i.e. with the terminal. The information to be transferred from both devices is related to commands and the initial image acquired by the sensor.

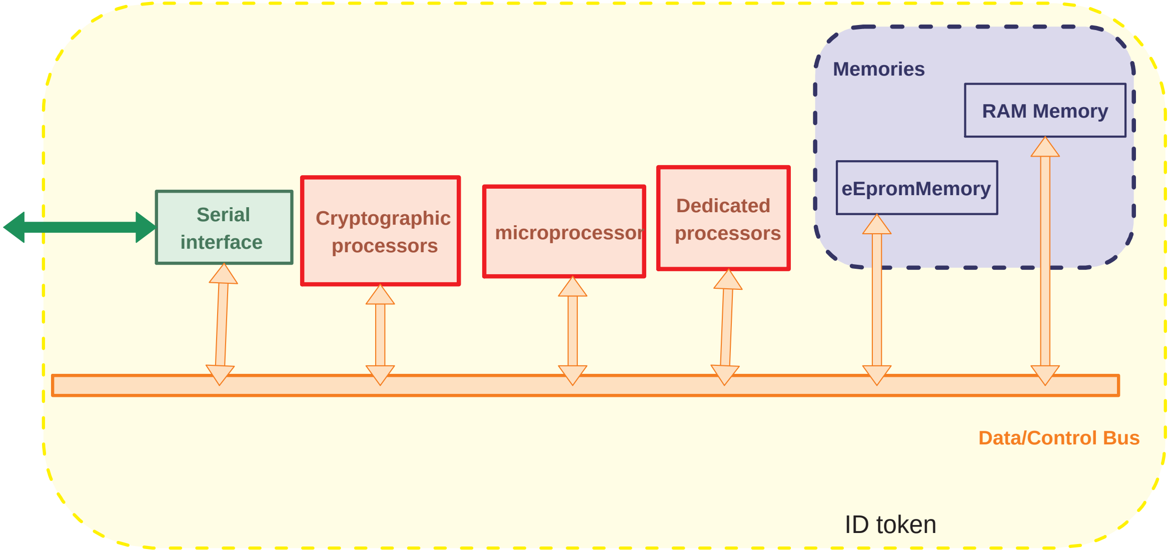


Figure 8.2: Architecture Proposed for an ID Token

The amount of data to be transferred depends on the image size and therefore, influences the choice of interface. A serial interface has been chosen for two main reasons: the first, due to the physical size of the interfaces, as other interfaces require a larger number of pins thus increasing the token size, and secondly due to the speed achieved by current serial ports, as transmission of large images do not require large amounts of time. Among the different choices of serial interfaces we highlight the use of the USB 2.0, a standard used for multimedia purposes which achieves a speed of 480 Mbits/s.

Cryptographic co-processor: All data sent or received to/from the token should be encrypted. Commands or any data transferral should be secured to avoid potential changes by intruders. Cryptographic algorithms such as AES or RSA, are recommended for use with the processor, as these algorithms have shown better results with respect to intruder attacks and are proven to be almost unbreakable.

Microprocessor: Due to the processing capabilities required to perform any biometric solution, it is necessary to include a microprocessor. Among the many reasons for including this element, is the necessity of a bus arbiter to access the different peripherals, particularly the memory and dataflow control.

Hardware co-processors: The requirement to perform biometric processes in the token increases the processing capabilities required within the token, thus the use of a low cost microprocessor would not be sufficient. The computational load of the biometric recognition algorithm in many cases is high enough to require special processors for specific tasks.

EEprom memory: This memory is necessary to store biometric data and other personal data which should be preserved even when no power supply is available. This memory should also be protected against possible physical attacks which may put the information it contains at risk; therefore this element must be tamper-proof.

RAM memory: This is used for all the temporary data received and sent by the terminal. This type of memory is necessary because direct processing of the external data received is not possible. The RAM memory should be erased after each use for two reasons: firstly, because of potential errors that occur from successive use due to old data and secondly, for security reasons, to avoid residual data.

Data and control bus: The amount of data and control in the token is not high enough for two different buses when considering the information transfer between modules. For this reason, only one bus has been included for this purpose. This bus, due to the information it transmits, should be highly secure and thus, implemented in an intermediate layer of the final layout.

Regarding security countermeasures, all such procedures have been considered in the previous chapter. During the fabrication process, the elements that form the token must be protected against physical and electronic attacks, i.e. restricting the access to information they store or process. Thus, by making the token tamper-proof, any attempt to access internal token elements will be noticeable; memories and other elements have been protected using glue and epoxy encapsulation techniques and their parts have been hidden, etc.

8.4 System functionality

The complete system performance is as follows:

The enrolment process: This process is considered as being performed in a secure environment which is controlled by an authorized agent. Once the template is obtained it is sent to the token. This accessing process to the EEprom memory and data storage requires high level security access; for this reason, the token has to be configured considering the aforementioned premises. During this process, the participation of an authorized agent is recommended to instruct the user during the enrolment process and to explain how the system will work for future accesses; this avoids the user having future problems due to forgeries, robberies or token loss.

The verification process: The verification process can be carried out automatically with/without supervision. The user, at this point, knows how to use the system and what is required for the interaction. For this reason, the user, when accessing the particular facilities must connect the token to the terminal, after which the procedure is as follows:

1. The terminal acquires the image and encrypts it using the public key.
2. The terminal sends this image to the token
3. The token receives the encrypted image and stores it in its RAM memory
4. The token decrypts this image using the private key stored in the EEprom memory.
5. The token performs the recognition process using the biometric algorithm implemented in the hardware and software.
6. If the result is an affirmative identification, confirmation is sent to the terminal in the form of an encrypted message using the private key. If the result is a negative identification, a different message is sent using the secret key. With this solution, the terminal is capable of detecting if the user is who he/she claims to be, and also if the terminal is the one expected.
7. In case of an affirmative identification, the terminal will communicate this to the central system where it provides the services the user has requested or the services the user is allowed to access. Finally, the RAM memory in the token is erased preventing further access to temporary biometric data.
8. If a negative identification occurs, the terminal registers it, but also provides a second or even third opportunity for the user to access the system by repeating the identification process. If after these two additional attempts the user is still not verified by the system, the terminal will block the token for future use, informing the central system of this action. For future access attempts the user must contact the system administrator to unblock the token.

Changing the token/Updating the token: In some cases, changing or updating the token is required. Among the many different cases are those related to possible token robbery, key annulment and renewal. In these situations, the user must return, whenever possible, the token to the system administrator to be changed for a new user public and private key without changing the biometric data; as it is not recommendable to change the biometric data. By combining cryptographic keys the security is increased and at the same time, compromising the biometric data is avoided. As regards the token theft case, only the keys should be changed.

With respect to the process to be performed, the token must be capable of carrying out the biometric recognition process which, in many cases, requires a high computational load. For this reason a hardware/software architecture has been chosen. With this type of architecture, processes which require large amounts of time using software can be implemented in hardware, also using hardware can speed up the process as a result of concurrence. Additionally, hardware provides other advantages when compared to software, especially in terms of security. The majority of software can be changed by upgrading the device firmware. In many situations, this is a beneficial advantage as the device can be changed and/or its functionality improved. However, for the current application, this particular device characteristic becomes an important weak point as attacks provoked by trojan horses or any other malicious programs can change the functionality of the system and therefore its response. For the same reasons the reconfiguration of the hardware is also unadvisable, i.e. the token should be permanent from the beginning where no possible changes can be carried out on its performance. However, as will be presented in the chapter on future work, a possible technique to change the functionality is by using a digitally signed encrypted code, where this can be used for both the hardware and software.

8.5 Hardware/software co-design for biometric algorithms

To distribute the biometric processes into hardware and software, the cost function proposed is the following:

$$F = c_a * \sum_{i=0}^{n} Area_i + c_t * \sum_{i=0}^{n} Time_i + c_p * \sum_{i=0}^{n} Power_i + \tag{8.1}$$
$$c_s * \sum_{i=0}^{n} Security_i + c_p * \sum_{i=0}^{n} Performance_i$$

In this formula, not only the nodes obtained from the data flow but also the connections and transferral between them are considered as vulnerable points of the tokens performance. As can be seen, several terms have been included in this formula. The term $c_t * \sum_{i=0}^{n} Time_i$ refers to the processing time expected for each configuration at each node and the transmission of data between them. The term $c_a * \sum_{i=0}^{n} Area_i$ is related to the occupied hardware and software area, the term $c_p * \sum_{i=0}^{n} Power_i$ to the power consumption. The term $c_p * \sum_{i=0}^{n} Performance_i$ is related to the differences between performing nodes in hardware and software and finally the term $c_s * \sum_{i=0}^{n} Security_i$ is related to the security associated with each module and transmission. Each of these summations is subject to parameters which are based on the designer's experience. These parameters allow the designer to emphasize those relevant terms for the specific system he/she is dealing with.

In the following subsections the different terms in formula 8.1 above will be described, where the meaning and necessity of each one will be pointed out, also described here are the techniques behind the approaches required to measure the different terms in the design process. Each of these terms not only represents values according to the processes parameters obtained in their correspondent implementations on hardware and software, but also, the connections between them have been considered, the bus transmission and the requirements for the necessity of external memories.

8.5.1 Time term

In accordance with the reasoning that most researchers have applied, the current proposal has included the time as an important factor can be improved by hardware. Processes can be performed on either platform, i.e. software or hardware, but several of the processes can be performed in less time when using hardware, especially those related to recursive and/or parallel computing. The acceleration obtained when using hardware is noticeable only in those processes, although any of them can be computed in hardware.

In this case, the time is measured and compared using time units, as both platforms are capable of returning the same parameter. Additionally, the time to transmit the data from one module to another has been considered, especially between modules on different platforms. During the time measurement, it is important to consider the scheduling process, i.e. the order of process execution and each start-up time. Here we identify three different types of processes:

Independent process: These processes do not require any data or control to commence computation.

Control dependant processes: These processes are combined with others because of the control they exert on them. They are required to start a control signal provided by other processes.

Data dependant processes: These processes require data from other processes to start. This is the case for the majority of processes in digital signal processing. These can be divided into two groups:

Memory dependant processes: When the amount of data necessary is excessively large and an external memory is required. The first process stores the data in a memory which will be accessed by the next process to acquire it. As an example, this is the case for processes related to image computations, where the size of the image is large thus it becomes unsuitable to transfer the complete image from one process to another and is transmitted and stored in auxiliary memories.

Non-memory dependant processes: If the data required consists of only a few parameters or any other values which can be transferred from one process to another, this removes the requirements for external memories during transfer processes.

These data dependencies should be considered in the computing time. In the case of memory dependant processes, the access to the memory should be considered both for the initial process and the following one, i.e. one for writing and the other for reading. We will consider these periods of time in each of the processing schedules, without taking into account if one is performed in hardware and the other in software. However, when processes are non-memory dependant, the platform used for both processes should be considered; when the platforms used are different, the data transmission on the bus requires a certain amount of time which should be taken into account according to the number of transmissions on the bus. If both processes are performed using software, the data transmission is not considered to be the same as that of writing or reading to an external memory. Finally, if both processes are performed using hardware and non-external memory is used, intermediate registers are required to store the transmitted data.

Another important consideration that must be taken into account at this point is the necessity to control the start of each process and to establish an execution order. This order is determined by the dataflow, however when two processes start simultaneously, particular criteria should be established to decide when each one of the processes can begin, such as:

- No two processes can be performed in software at the same time as it is not possible for the microprocessor to carry out parallel computation.

- Two processes can start at the same time when one is performed on hardware and the second using software, or when both processes are built into hardware and are not memory dependant.

- When two processes can be performed at the same time and both are data dependant, it should be considered that both may attempt to access the external memory simultaneously. In this case, the data will be multiplexed if the access is in the same order. However, situations may arise where the access required by the simultaneous processes should be performed in a different order than that established. In this case the following protocol is considered:
  - If the security related to one of the processes is higher than the other, this process should start after the former.
  - If the same security level is required for both processes, the process that requires more time should access the memory first, this is done in case both processes access the memory at the beginning of their performance.
  - In the case where both processes require the same security level and require continuous access to the memory, the order of execution is unspecified.
  - If a process is data dependant on two other processes, these two processes should start so that the data is available for the other process, i.e. in this way the waiting time for the data is reduced in the memory to avoid potential attacks.

This scheduling algorithm does not affect the systems performance. However, the working environment has been considered at all times, especially when related to the risks these systems are subject to with respect to potential intrusions, this is achieved by reducing the time the data is stored in the memories.

8.5.2 Space term

Due to the portability requirement imposed on these systems, the space should be minimal to facilitate their insertion in either small tokens or bigger systems, such as personal computers. The space is determined by the term: $c_a * \sum_{i=0}^{n} Area_i$ When measuring the space, we have encountered a problem which is related to the difference between the software and hardware. In software, the space occupied is measured in terms of lines of code, and therefore, their occupancy in the ROM memory where it is stored. In hardware, the space is measured in terms of the number of elements used, and when the system is developed using an FPGA by the number of flip-flops and the LUT used, this is translated into cell block occupancy. To combine both solutions, we propose the use of percentages of both elements. In the software case, we have considered the percentage of code a process requires, the same is applied to the amount of components required by the hardware. In these considerations, the limitations of both platforms should also be accessed:

$$\sum_{i=0}^{HW} A_i \leq 100\% \tag{8.2}$$

$$\sum_{i=0}^{SW} A_i \leq 100\% \tag{8.3}$$

Both software and hardware platforms cannot occupy more space than that limited by the size of the devices, i.e. the complete EEprom memory and FPGA occupancy. All these assumptions cannot be considered when designing an ID token using the SoC solution. The System on Chip solution allows the inclusion of the microprocessor and the dedicated co-processors on the same chip. In several SoC's, the microprocessor is embedded (hard core processor) and in others the dedicated hardware is configured to create a microprocessor (soft processor). When working with soft processors and SoC solutions, the total area can be measured by considering the logic blocks, as both solutions rely on the same blocks. Even though the measurement is more straightforward; it must be considered that when computing the area restriction, the microprocessor itself requires several logic blocks and therefore, the area occupied by all the elements should be inferior to the complete chip. This area is determined by the dedicated co-processors, the soft microprocessor and the code this performs.

8.5.3 Power consumption

ID tokens, as portable devices, require an external supply to operate. The requirements of this particular parameter differ from one architecture to another, and it is directly related to the power supply available from the terminal. In the assumptions presented in this Thesis, this parameter has been considered as a further parameter in the partition function. As regards power measurements, both platforms are measured and compared using the same units: mW. However, not only should the power consumed when a process is being performed in a platform be considered but also the following should be taken into account:

- When the microprocessor is in the idle state, it still consumes a certain amount of power; this data can be obtained from the manufacturer's datasheet.
- A similar situation occurs with the hardware, i.e. all the processes performed in the hardware consume power whether they are active or not. This consumption is called static or quiescent power and is caused primarily by the leakage currents in the transistors which form the hardware.

8.5.4 Security term

When considering security, the hardware provides additional advantages when compared to software solutions. The risk of an attack is higher in software-only based solutions. Details regarding the majority of attacks on tokens have been presented in chapter 4 which refers to attacks on software solutions rather than those in hardware. The changing of firmware is only possible in software, hardware is considered as something unmodified. Moreover, the attack on the memories is not so complicated when performed using software, as the specific drivers are readily available on the worldwide web. Attacks on hardware processes are possible using techniques based on reverse engineering. However, considering the physical requirements, described in the previous chapter, these are difficult to perform as the intruder requires direct access to the token. For this reason, and considering the aforementioned attacks, a security term has been introduced which provides information on which platform each process is performed.

When the security related to a process is required to be high, the process should, if possible, be performed using the hardware, thus reducing the potential attacks on this process. In cases where the security related to a process is not required to be high, they can be included in the software. For example, when considering the matching algorithm; it is highly recommended that this process is performed using hardware, as amongst all processes, it is the most vulnerable part to attacks as it makes the decision on the identity being examined.

Also in terms of security, we should consider the communication bus between the software and hardware.

Among the most common attacks in authentication protocols are those derived from eavesdropping and man-in-the-middle. These attacks attempt to acquire the information and later, using techniques such as hill-climbing or replay-attacks, the intruder can impersonate the authorized user. To prevent these attacks, the busses and the information they transmit should be placed under surveillance. For this reason, it has been decided to include, within the security term, the communication between the different processes. By including this term, the low security communications between the hardware and software, and those where the information is most suitable for being replicated under two hardware co-processors is left out. Using this solution, the communication between the platforms and the use of external memories is avoided thus reducing any risks.

8.5.5 Performance term

Finally, the performance term has been added to ensure the algorithm functionality. When developing systems based on hardware/software co-design, problems due to truncation were provoked by fixed point operation in the hardware. Most hardware implementations use this fixed point method as opposed to software which use floating point operations. The difference between these operations is in the representation method for radical numbers: the first one uses a fixed number of bits to represent the number after the comma, the second represents the numbers using two types of data, the radix and the exponent, and therefore, allows a wider range of numbers to be represented.

Although most researchers recommend using fixed point operations in both platforms, here it has been decided to use floating point arithmetic so that the software does not suffer any loss of accuracy. In software, due to memory management, the data always takes on fixed length values according to the variable declaration: 8, 16, 32 or 64 bit lengths. Working in hardware with this floating point arithmetic increases the hardware area and design difficulty. For this reason, it is recommended to use a shorter length and fixed point arithmetic where the minimum amount of accuracy is lost. This data length reduction and its arithmetic may lead to errors, where this is translated into reduced performance and therefore misidentification errors.

For this reason, a term which considers the variations in hardware provoked by this length truncation has been included in the partition function. This term will be null when the process is performed using software, but it will take on the value of the variance of the difference between the hardware and software in cases where the processes are implemented in hardware. This variance is obtained by computing the difference between the hardware and software implementations of the same process and determining the mean deviation of the resulting data.

8.5.6 Solving the partition problem

As has been mentioned in previous chapters, minimizing the partition problem requires the use of special techniques. In the work presented in this Thesis the three aforementioned methods are modified slightly to obtain the minimum of the cost function according to the requirements of the ID token.

In all the approximations presented, it has been attempted to reuse the hardware areas whenever possible. In digital signal processing, many methods used are quite similar, as they are based on the same mathematical analysis. For example, when considering equalization which is based on the values from the histograms. If we try to stretch the histogram to use all the possible histogram values, not all pixels will be located in the same area. This process is comparable to that used to eliminate the pixels below a certain value. In both cases, the histogram must be computed as well as specific criteria established to change the pixel values. Thus, in hardware or software, the implementation of both methods relies on the same module but for different parameters.

Additionally, due to the platforms power consumption, even when it is idle, an attempt has been made to have the platform continually processing. In some cases, both the processor and the hardware can work at the same time, however, this can be modified according to the designers criteria. In several situations where this is not possible a process execution order should be established.

8.5.6.1 Simulated annealing

Simulated annealing has been based on the modification of a parameter known as temperature as the expected minimum is approached or until a number of iterations have been performed. This value decreases in each of the iterations where the set examined provides better cost results than the best solution described thus far and where the energy, $e^{(\triangle C/T)}$, of the system between two neighbouring sets is below 1.

As has been previously mentioned, this method presents difficulties when locating a minimum positioned far from the initial set examined, this is particularly true when the cost function has several local minimums. This problem arises due to the algorithm itself, as when a minimum is reached, it is impossible to satisfy any of the previous requirements to keep searching in the solution space.

The algorithm has been modified as follows to consider this approach: Considering the problem related to the localion of the minimum that arises, an algorithm with a certain penalty has been implemented. Basically, this algorithm works in a similar way to the previous technique but if during the computation the energy found between the sample considered and the neighbour is below '0', the temperature value is updated by increasing it slightly to show that the energy obtained is not as good as that previously obtained value. This solution allows movement within the space of solutions into others which differ from the initial solution.

8.5.6.2 Tabu search

In the tabu search, most contributions made are related to the tabu list. We have introduced a long term tabu list, which contains forbidden movements. This list is formed by using movements chosen by the designer and according to the final requirements of the system. For example, if a highly secure device is being designed, the matching process is always forbidden to be performed using software. The sets contained in this list are fixed, and therefore, are not modified during the minimum search process. However, the frequency list, as mentioned in [55], is that of diversification of the results, as it is formed from several sets from different parts of the possible solution space. As a result, we have been able to move from one region to another to find new potential minimums.

In contrast, the short term tabu list is formed by the sets already explored, where these were removed from this list after a number of iterations. The length of this list is fixed to 5 elements, as the number of possible nodes being dealt with in this case is not large. However, if an algorithm is examined and it leads to several nodes, it is recommended that this value is increased.

The aspiration criteria have been fixed according to the requirements of the ID token designed. Considering again the case of designing an ID token, where the main concern is security, aspiration criteria is considered as any solution which increases significantly the security of the ID token.

8.5.6.3 Genetic algorithms

Genetic algorithms rely on the modification of the population to obtain improved results. In this case, no contributions to this method have been carried out.

We have employed the cost function as the fitness function. This information has been used in the crossover probability. These probabilities are computed using the fitness function; where higher values are assigned to those with a lower cost function and lower ones to those sets with a high cost function. By applying this solution, the probability of changing those sets with lower cost functions is increased. This solution is known as the roulette approach [67].

In this chapter we have presented a new methodology for the development of a recognition system based on ID tokens. We have focused the development of the design on a hardware/software co-design. Our proposal attempts to consider the different aspects of the working environment and requirements of ID tokens such as the processing time or area, but also some other aspects which are generally not considered by other designers such as power consumption and security, features which can be improved by using effective implementations of a hardware or software module. For the proposal presented in this Thesis we have defined a new partition function which considers all these terms and has led to an effective configuration when concerned with the system requirements. Among the different terms considered we have highlighted issues of security, which compels the development of modules using hardware as they are more suitable for protection against attacks. The term related with the processing time has also been presented, here a scheduling algorithm is employed which also attempts to improve security without any loss of time. Investigation of the accuracy term has led to reduced error rates which may occur in the final configuration, this avoids misidentifications when using the ID token in the final system.

Chapter 9

Modules for Iris ID token

To prove the feasibility of the methodology proposed in this Thesis, we have implemented an ID token which considers an application based on Iris Biometrics has been implemented. The work presented in this Thesis has been performed in the following steps: first, the iris algorithm has been studied with the purpose of obtaining the dataflow and its corresponding graph, thus dividing the algorithm into different processes. These processes have been implemented using both hardware and software, where different implementations have been considered for each of them. A study of all these implementations has provided the essential design data such as processing time, area, etc. required for the co-design phase. Therefore, the implementation of all the iris algorithm modules on both hardware and software has been the first step of the design process.

9.1 Developing tools

To implement the ID token a SoC device has been used. The SoC solution allows the processor and the dedicated co-processors to be included on the same chip. These solutions do not demonstrate the magnitude of our proposal as both hardware and software are contained within the same device; i.e. any communications are not carried out on external buses, also the security of implementing a process in hardware or software cannot be appreciated. However, the use of this type of solution for the implementations presented in this Thesis has been motivated by two different reasons:

- Manufacturers provide tools which make the development and debugging of the system relatively straightforward.

- Implementing these systems on a SoC is the first step taken to prove the combination of a hardware/software architecture, as we pointed out in chapter 6

The Xilinx embedded solution has be selected for this research. The solution proposed by Xilinx depends on the FPGA considered. Xilinx's manufacturer has proposed the use of three microprocessors for their FPGAs:

- PowerPC Hard Processor: The IBM PowerPC 405 core is a hard 32-bit RISC CPU core embedded directly within the Xilinx FPGA and is used to implement high performance embedded applications. The combination of dual hard PowerPC core systems integrated along with co-processing capability enables a wide range of performance optimization options. The PowerPC integrates a scalar 5-stage pipeline, separate instruction and data caches, a JTAG port, trace FIFO, multiple timers and a memory management unit (MMU).

- MicroBlaze Soft Processor: The MicroBlaze core is a flexible 32-bit Harvard RISC architecture with a large instruction set optimized for embedded applications in Xilinx FPGAs. One of the main features of the MicroBlaze architecture is the control it has over cache sizes, interfaces, and execution units. The soft processor nature of MicroBlaze makes it very customizable. A trade-off between the different features and size can be made to meet price and performance goals. With the MicroBlaze soft processor, no royalties need to be paid and the processing solutions developed never become obsolete. MicroBlaze even integrates a low-latency IEEE-754 compatible Floating Point Unit (FPU).

- PicoBlaze Soft Processor: PicoBlaze is a compact and cost effective fully embedded 8-bit microcontroller core. PicoBlaze is provided as a source-level VHDL file with royalty-free use in Xilinx FPGAs and CPLDS, thus is immune to product obsolescence. This core requires no external resources; it is totally embedded within the FPGA. Because of its small size, applications can implement multiple PicoBlaze instances to address simple or sophisticated tasks.

From these options the one chosen for the implementation has been the Spartan3-SX3C2000 from the SPARTAN 3 family. The Spartan3 family of FPGAs is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. Because of their exceptionally low cost, Spartan-3 FPGAs are ideally suited for a wide range of consumer electronics applications including broadband access, home networking, display/projection and digital television equipment. This family is not provided with a hard core embedded microprocessor, reducing the selection of microprocessor to two possible alternatives: Microblaze and Picoblaze. We have used the Microblaze solution as the Picoblaze is an 8-bit microprocessor and the algorithm performance may be inferior to that expected, increasing the error rate.

Two different buses can be connected to this microprocessor:

- Fast simplex link (FSL): This bus is similar to a direct connection between the hardware co-processor and the Microblaze, where no protocol is followed. This link only accesses the memory direction of the Microblaze memory map, before the hardware starts its computation. This connection is relatively simple and therefore, provides the fastest connection possible as no protocol loads are added.

- On-Chip peripheral bus: The on-chip peripheral bus (OPB) is designed for easy connection of on-chip peripheral devices. It provides a common design point for various on-chip peripherals. The OPB is a fully synchronous bus which functions independently at a separate level to the bus hierarchy. This bus has been used in this Thesis to connect the different co-processors to the Microblaze.

For these FPGAs, Xilinx offers a wide range of tools to develop different applications. In this Thesis several tools have been used:

- Modelsim: although Xilinx is not the manufacturer of this tool, it has been used as it is one of the most widespread tools for simulation purposes. It allows the creation of VHDL files, compilation and simulation.

- Xilinx ISE: The main Xilinx application allows the integration of several tools. It has been used for synthesizing the VHDL modules.

- XPower: Xilinx have provided this tool for the Xilinx ISE to compute the power consumption of the different designs. This tool provides information on the power consumption when the module is running and also when it is in the static state (also known as quiescent). To use this tool, several steps should be performed first: The design should be simulated and synthesized. Once this has been done, a new test bench should be created for power consuming purposes. This test bench leads to a file which examines all the possible changes and variations in all of the signals of the process, and which, accompanied by the synthesized design, is used by the Xpower tool to compute the power consumption of the design. Due to the tedious process and the size of the files, Xilinx have provided a more straightforward tool based on Xpower and Microsoft Excel, which only requires the synthesized design. This tool has been used, as performing all the processes described above requires high performance machines and extensive computational time.

- EDK: This tool has been used to create the complete system. It permits working with hardware/software co-designs by embedding modules developed in dedicated hardware and modules using software. Additionally, this tool provides drivers for different peripherals which are located on the platform board, including our own peripherals.
- Matlab: Used for computing the first steps of the design, and to prove the viability of the algorithm.
- Borland C++: In the same way as previous tools, it is also used to test the algorithm and to study the co-design partition problem with the data obtained from the hardware and software modules.
- System Generator for Matlab: This library is provided by Xilinx for Matlab Simulink and allows hardware simulation to be performed on the modules designed in hardware and also provides several elements which are typically used in hardware design, therefore, allows modules to be designed. This tool also calls the ISE to synthesize the modules developed as part of the EDK co-processors. Additionally, it allows previously designed modules to be introduced into VHDL as black boxes for bigger designs.

Fig. 9.1 shows the relation between the different tools employed in this Thesis. Matlab, Simulink and Xilinx System Generator have aided us in developing algorithm prototypes in hardware and also to verify them on the board. The software modules have been developed using Xilinx EDK, this has allowed us to embed a microprocessor on the FPGA and to program it. To simulate these modules, we have used Modelsim. Once all the modules have been implemented and simulated separately, Xilinx ISE has then been used for the hardware and EDK for software modules and allows data such as the area occupied and the processing time to be obtained. The power consumption data is obtained using an excel tool based on Xpower.

9.2 Encrypting/Decrypting module

In the approach followed in this Thesis, we have used a cryptographic co-processor based on AES. This co-processor was obtained from [65] [21].

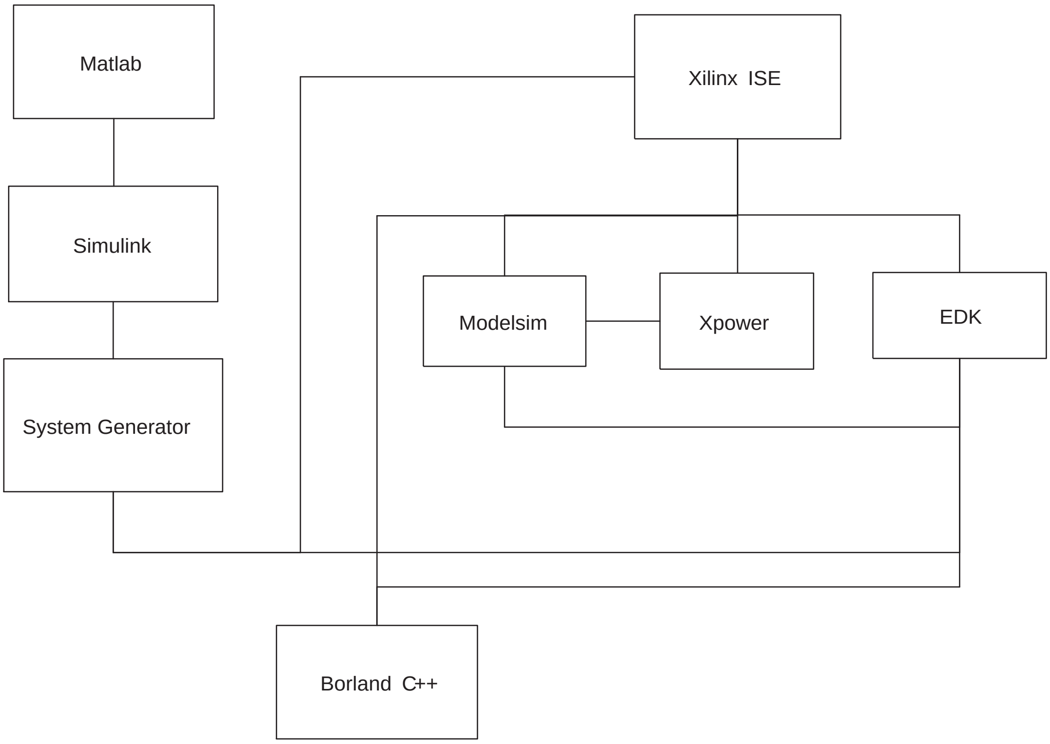


Figure 9.1: Tools used in this Thesis

The AES protocol has been used to encrypt the data from the token transmitted to the terminal and vice versa. This protocol was proposed by Rijndael in 2001. AES is based on a design principle known as Substitution Permutation Network. It is fast for both software and hardware, relatively easy to implement, and requires little memory. The standard AES definition has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits; the fixed block size of 128 bits is $128 \div 8 = 16$ bytes. AES operates on a 44 array of bytes, termed the state. The AES ciphering is specified as a number of repetitions of transformation rounds that convert the input plain-text into the final output of the cipher-text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform the cipher-text back into the original plain-text using the same encryption key. The algorithm works as follows:

1. Addroundkey
2. Rounds: the number of rounds to perform depends on the key length, from 14 rounds when using keys of 256 bits, to 10 rounds for 128 bit key lengths.

   (a) Subbytes: a non-linear substitution step where each byte is replaced with another according to a lookup table.
   (b) Shiftrows: a transposition step where each row of the state is shifted cyclically a certain number of steps.
   (c) Mixcolumns: a mixing operation which operates on the columns of the state, combining the four bytes in each column
   (d) Addroundkey: each byte of the state is combined with the round key; each round key is derived from the cipher key using a key schedule.

3. Final round (no Mixcolumns)
   (a) Subbytes
   (b) Shiftrows
   (c) Addroundkey

The implementation of this module in the token has been done using hardware for one important reason: The vulnerability of this module is quite high when considering the complete system process. This module receives the data and the commands from the terminal in this master-slave architecture. Therefore, possible modifications to the data received or during the decrypting process can lead to errors in the process; moreover, also avoids attacks on the system performance due to hill-climbing attacks. The architecture of the block used is presented in Fig. 9.2:

This module obtains the following data:

- Hardware area:
  - Look-Up table (LUTs): 17334 (42.32%)
  - Flip-flops: 1739 (4.25%)
  - Slices: 8998 (43.94%)
- Time:
  - Maximum Frequency: 50.702 MHz
  - Total processing time:
    - For each command (32 bits): 256.4001 ns
    - For the complete image: 1.231 ms

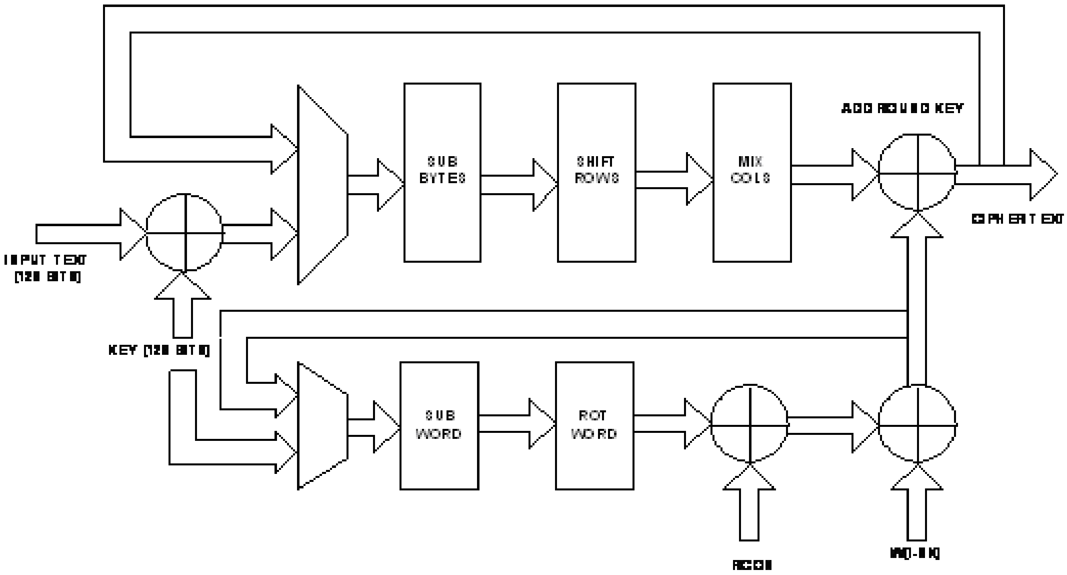


Figure 9.2: AES hardware implementation

- Power:
  - Static: 0.187 mW
  - Dynamic: 0.016 mW
  - Total: 0.203 mW

9.3 Iris algorithm modules

The algorithm used to design the ID token has been the base-line algorithm which was described in chapter 3. This algorithm was provided by the research group this Thesis has been developed within, and although it does not provide improved performance results when compared to other algorithms, its complexity allows the verification of the proposals feasibility. An in-depth analysis of this algorithm has been presented in the aforementioned chapter, here the dataflow obtained will be presented (Fig. 9.3).

This dataflow, as it can be observed, is clearly divided into two main groups, those processes which compute the pre-processing and those which control the feature extraction block and comparison. The first group requires continuous memory access and therefore, the possibility of concurrence is reduced by these memory accesses. Fig 9.4 presents a detailed description of the pre-processing algorithm. In this figure, the dependency between the different processes is also indicated: the processes which are

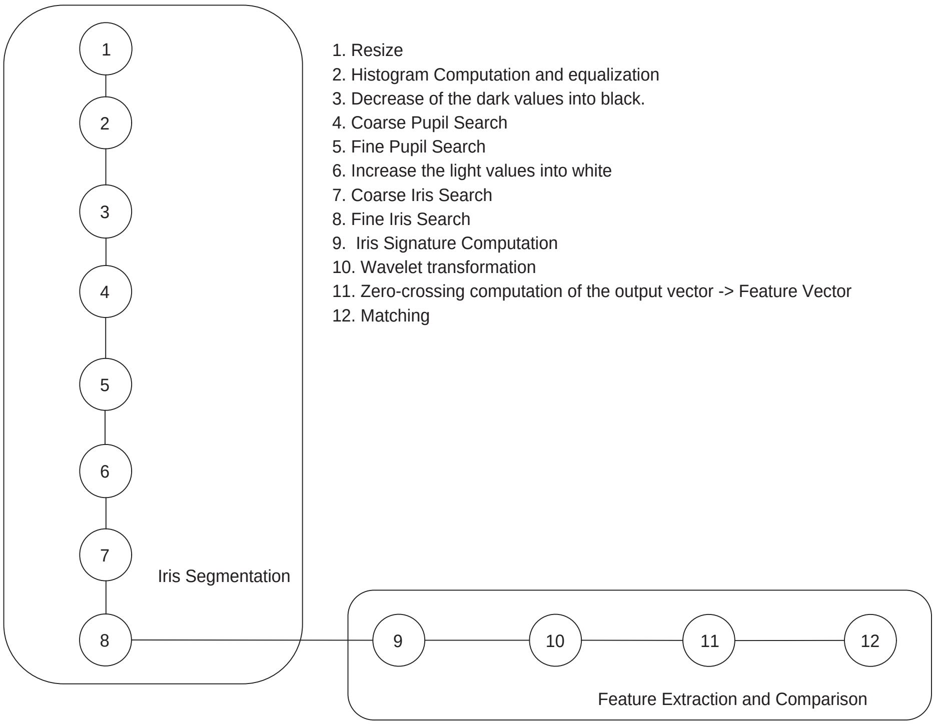

1
2
3
4
5
6
7
8
9
10
11
12
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching
Iris Segmentation
Feature Extraction and Comparison

Figure 9.3: Algorithm Data Flow

external memory data-dependant are indicated in red and in blue are those which store the resulting data in a memory and finally, in green are those which are data dependant, but do not require external memories. Several processes, as it can be observed, are both external and non-external data dependant. Different RAM memories have been named in this figure, although all of these are the same. In the figure above, we have included different blocks which refer to RAM memories and indicate the changes in the data stored in the memory, and which data is required by each module. Some modules will write in the RAM memory thus change the stored data and several others only require the memory to be read where the data is not modified.

As can be observed form Fig. 9.4, the processes in the pre-processing block are all data-dependant. Several of these require external memory to read and write the results obtained; others only require the data from previous processes. Within the first group are those related to the initial pupil or iris search, which scan the image in order to find variations caused by these elements. In the other group, the processes related to the fine search or parameter determination may be found.

The block related to the feature extraction block and the matching process, due to the amount of data it requires, does not require memory access as the data studied is stored using only 256 bytes, i.e. the iris signature. This is the only module that

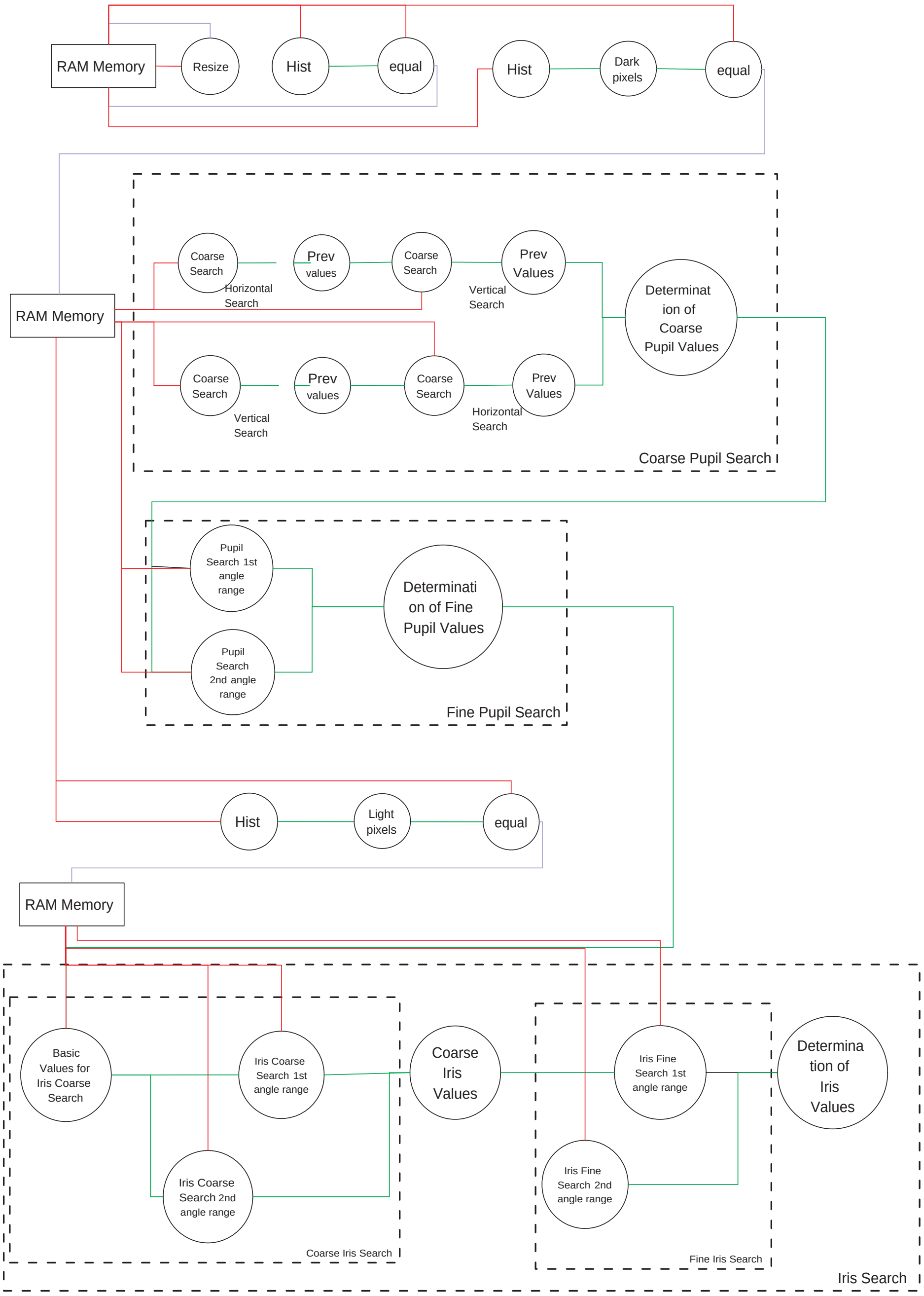


Figure 9.4: Detailed Pre-Processing Data Flow

requires memory access to compute these values.

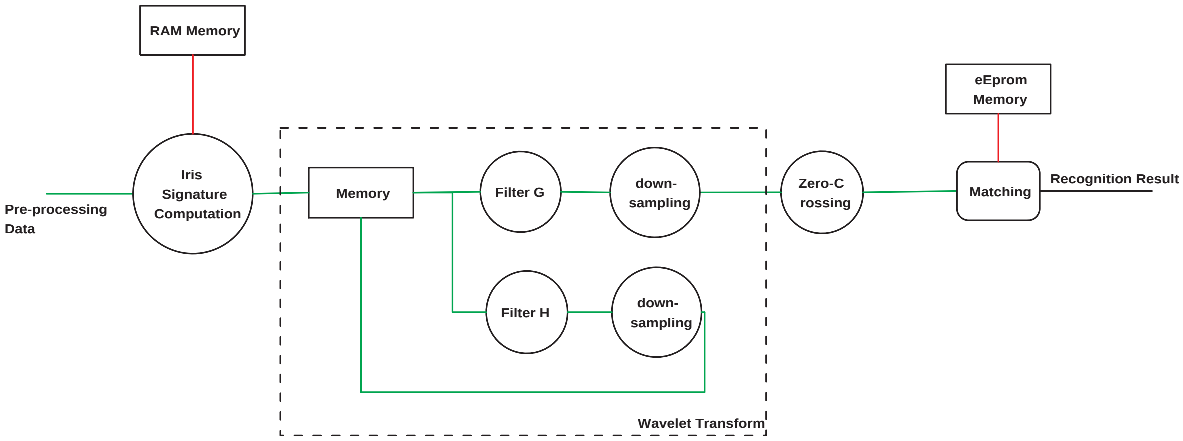


Figure 9.5: Feature extraction and comparison Data Flow

The dependency between processes is clearly based on the data, although external memories are not required. The values transferred from one process to another are of 256 words; the size of this word depends on the filter coefficient parameters and the input values, this will be observed in the hardware implementation.

Regarding the wavelet transform, it is important to point out that several scales have been used that can be computed according to wavelet theory. The number of scales depends on the information required, as the wavelet and its zero-crossing provide information in function of the scale that has been considered. Thus, when looking for sharp differences, the information will be obtained in the first scales and so on.

The comparison block is data dependant but not memory dependant. The matching algorithm reads the EEprom memory, which is only read by this block, and as a result there is no waiting time for other processes required by this memory. This process is also data dependant with the zero-crossing block which provides the feature vector.

Considering Mallat's fast wavelet transform computation and the output latency of this block, the feature extraction block and the matching algorithm can operate in parallel. In each time slot both filters can work concurrently, the previous results of the filter G can be computed in same slot as the zero-crossing representation. At the same time, the previous zero-crossing representation is computed by the comparison block. This algorithm avoids time being wasted when obtaining the complete feature vector required for comparison.

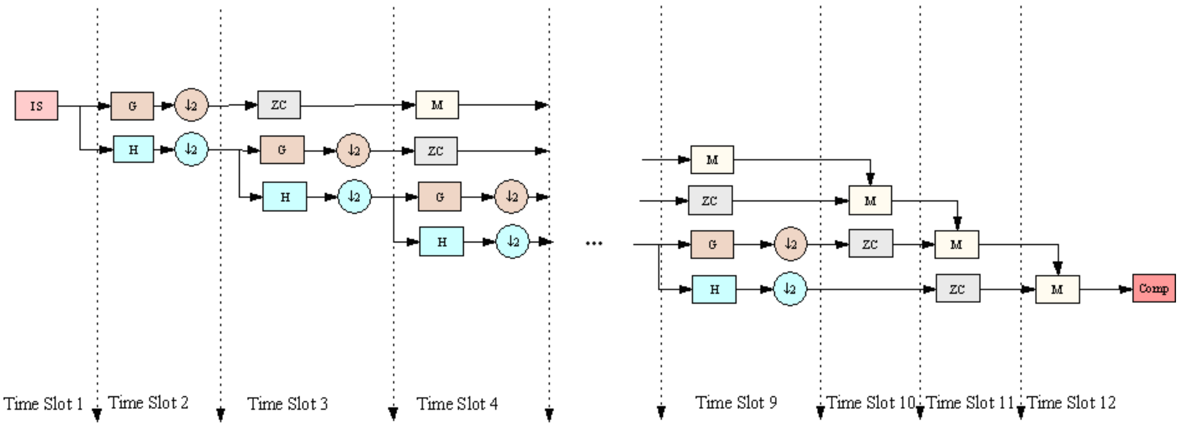


Figure 9.6: Feature extraction and comparison concurrence

9.4 Software implementation

The software implementation has been carried out using standard C, as no further languages can be implemented using the EDK. The code implemented for the microprocessor is stored in several blocks known as block RAMs, these are situated at the FPGA boundaries. When designing the general platform, the designer should specify the space within each block RAM that is dedicated to storing code. The code size is quoted as the number of code lines, but not those which correspond to the occupancy of the block RAMs as it is a space reserved for storing code. The EDK provides the possibility of using three different size areas for this purpose: 8, 16 or 32 KB. We have chosen 16 KB as it is the minimum size possible required for the complete algorithm code to be included.

The microprocessor, apart from computing part of the biometric process, controls other functions, these are:

- Controls the complete process: The microprocessor is in charge of activating and deactivating the modules required for each action.
- Controls the bus access: Possible collisions can occur when the token is working, especially when accessing peripherals. The microprocessor is in charge of each process to access the bus avoiding collisions.
- Input/output communications: The microprocessor is in charge of controlling communications with the outer world, i.e. storing the initial data received from the terminal, calling the encrypting module to decrypt data, providing results, etc.

Table 9.1: Time results for software modules

	Size		Processing Time	
	Code lines	%	Cycles	total time (μs)
Resize	92	15.54	42291	1057.275
Equalizer	96	16.22	32288	807.2
Histogram	88	14.86	22290	557.25
Pupil Coarse Search	96	16.22	35656	891.4
Pupil Fine Search	88	14.86	36004	900.1
Iris Outer Boundary Search	112	18.92	24000	600
Iris Signature Computation	92	15.54	23060	576.5
Wavelet Transformation	100	16.89	42800	1070
Complete algorithm	592	100	258389	6,459.725

The results obtained from the Microblaze microprocessor are as follows:

- Hardware area:
  - Look-Up table (LUTs): 2407 (5.88%)
  - Flip-flops: 1421 (3.47%)
  - Slices: 1739 (8.49%)
  - BRAM: 8 (20%)
- Power:
  - Static: 0.187 mW
  - Dynamic: 0.034 mW
  - Total: 0.221 mW

The results obtained using software are summarized in table 9.1.

9.5 Hardware implementation

As mentioned above, all the modules have been implemented in hardware to obtain the necessary data for the cost function. These implementations have led us to several contributions within the hardware development field, as we have proposed new architectures for several different methods.

9.5.1 Basic modules

Some processes and operations are required on several occasions during the operation of the ID token. Some of them are not easily implemented on hardware, and so several approaches exist in the literature. In this subsection, details are provided on the blocks implemented for these operations which will be instantiated on several occasions in the hardware modules blocks. The basic modules implemented are a division module and two other modules related to changing data formats from floating point arithmetic to fixed point arithmetic and vice versa.

9.5.1.1 Division module

As Opposed to multiplication which is performed in most FPGAs using dedicated blocks, dividing numbers in hardware is not a trivial task, several approaches have been proposed in the literature to perform this operation according to the type of numbers being dealt with. In our algorithm, all the divisions which should be performed are related to integer numbers and the result is always rounded to another integer, as all instantiations of this module are done in finding addresses in the pre-processing block.

Several approaches can be found for the division operation in hardware, generally these are divided into two main groups: slow algorithms and fast algorithms. Slow division algorithms produce one digit of the final quotient per iteration. Examples of slow division include restoring, non-performing restoring, non-restoring, and SRT division. Fast division methods start with a close approximation of the final quotient and produce twice as many digits of the final quotient in each of the iterations. In hardware, the most common algorithms used are the slow methods, as the fast algorithms require more hardware area and initial approximations. Slow division methods are all based on a standard recurrence equation (Formula 9.1).

$$P_{j+1} = R \times P_j - Q_{n-(j+1)} \times D \tag{9.1}$$

where P_j is the partial remainder of the division, R the radix, $Q_{n-(j+1)}$ the digit of the quotient in the n-(j+1)position, where the digit positions are numbered from least-significant 0 to most significant $n-1$, n is the number of digits in the quotient and D the denominator.

The most widely used slow division algorithms are:

- The restoring algorithms, which operate on a fixed-point fractional number and depend on the following assumptions:

  - $N < D$
  - $0 < N, D < 1$

  The quotient digits Q are formed from the digit set 0,1.

- The non-restoring algorithm, this is similar to the previous algorithm except that the value of $2 * P_i$ is saved, so D does not need to be added back in for the case of $TP_i \leq 0$.

- SRT division: Named by its creators (Sweeney, Robertson, and Tocher), SRT division is a popular method for division in many microprocessor implementations. SRT division is similar to non-restoring division, but it uses a lookup table based on the dividend and the divisor to determine each quotient digit.

All these algorithms require at least as many cycles as the dividend length. We have implemented a non-restoring algorithm as it is one of the most straightforward while at the same time, it does not require large use of hardware area. It may be observed that we have attempted to make use of simple hardware elements. This implementation is set to a 16 bit long dividend, as in most cases this is the length of the dividend. Fig. 9.7 shows the division module.

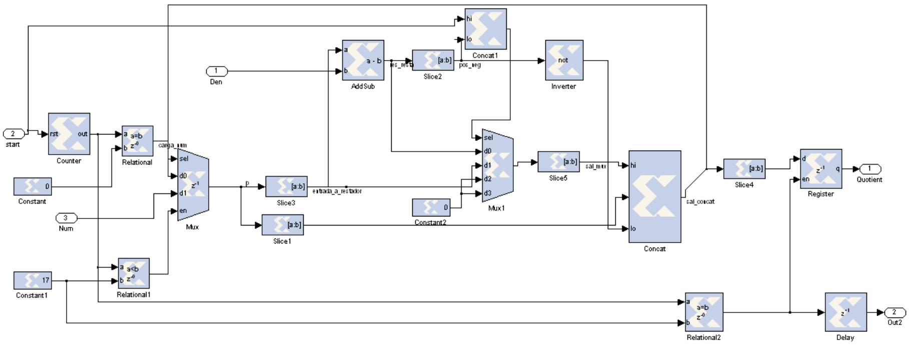


Figure 9.7: Hardware implementation of the division operation

9.5.1.2 Conversion of data format modules

Another important block that has been used several times is related to the data format change, i.e. from floating point to fixed point and vice versa. This block is only used when there is data transmission from the software to the hardware and for real numbers. This data, as previously mentioned, is formatted using floating point arithmetic when

Table 9.3: Floating Point number

S	exponent	radix
32	31-23	23-0

Table 9.4: fixed point fractional number

Integer part	fractional part

it is used in software, however, our implementations in hardware has been carried out using fixed point arithmetic. For this data format change, the implementation of these number formats must be considered:

- Floating point: numbers in floating point arithmetic are formed using 32 bits. These bits are organized as table 9.3 shows.

  The most significant bit is associated with the sign of the number, this is followed by 8 bits which are dedicated to the exponent and finally the fraction or radix occupies the remaining 23 bits. The exponent is defined in the standard IEEE 745 as the exponent of base 2 plus 127.

- The fixed point arithmetic is distributed as table 9.4 indicates.

  The number of bits used depends on the designer's criterion, as well as the decimal position chosen.

For this reason, the translation from floating point to fixed point is carried out with the block diagram shown in Fig. 9.8. The translation, as it can be observed, is performed by slicing the floating numbers into several parts and shifting the fractional part as many times as the exponent indicates.

An analogue block has been created to perform the transform from fixed points to floating points. The hardware results for these blocks are as follows:

- Hardware area:
  - Look-Up table (LUTs): 32 (0.08%)
  - Flip-flops: 12 (0.03%)
  - Slices: 30 (0.15%)
- Time:
  - Frequency: 167.392 MHz

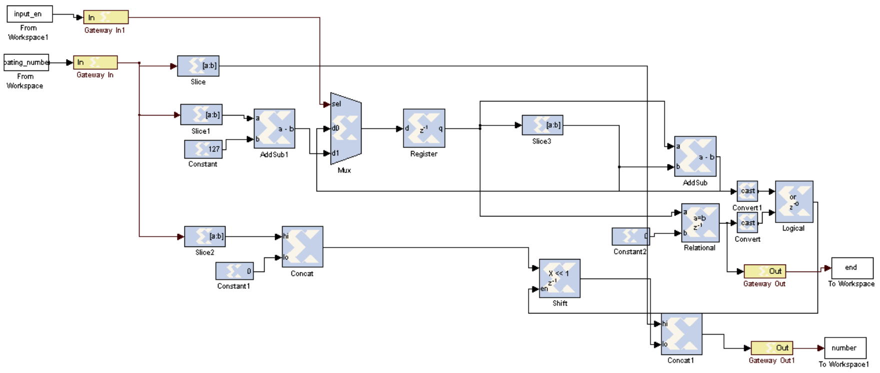


Figure 9.8: Conversion data format module

  - Total processing time: 0.102 ns

- Power:

  - Static: 0.187 mW
  - Dynamic: 0.014 mW
  - Total: 0.201 mW

The incorporation of this block will depend on the configuration under investigation and will be added only when it is required, i.e. when one block is performed in hardware and the following in software or vice versa, considering that any of these blocks work using floating point numbers and the data from either is transmitted via external memory or through the bus.

9.5.2 Pre-processing blocks

In the majority of cases these blocks require memory access, as the data required for computing their process is generally large enough so as not to be stored in the auxiliary memories. Due to this memory access, it is not possible to carry out simultaneous processes as memory accesses are sequential.

9.5.2.1 Resizing module

The first block to be performed is the resizing of the initial image into a smaller image. This resizing is performed for two reasons:

- A small image requires less memory space.

- Less time is required to locate the initial values of the pupil where the edge is sharper in a small image when compared to larger ones.

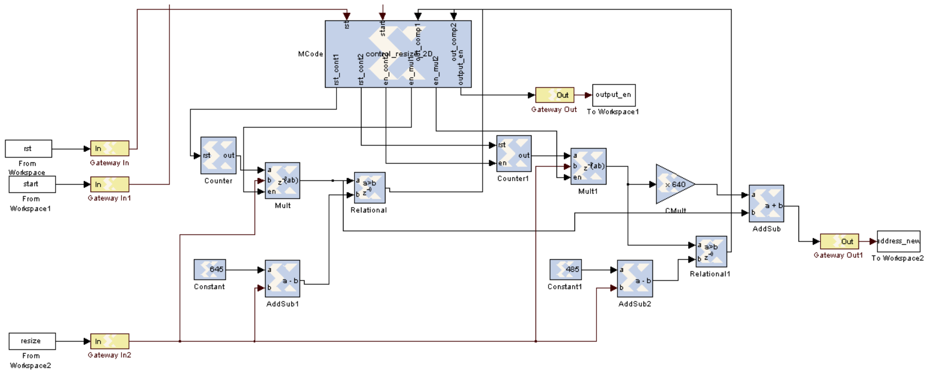


Figure 9.9: Hardware implementation of the resize module

The resizing process is implemented as Fig. 9.9 shows. This block is quite straightforward and is formed using counters for both the rows and columns, where all of these elements are controlled using a finite state machine. This module works by reading from the memory and writing the resized image in a different address section. The control block considers the memory delays in both reading and writing addresses. When measuring the data required for the partition process, the results obtained for this block are the following:

- Hardware area:
  - Look-Up table (LUTs): 312 (0.76%)
  - Flip-flops: 222 (0.54%)
  - Slices: 183 (0.89%)
- Time:
  - Frequency: 109.386 MHz
  - Total processing time: 112.391748 ns
- Power:
  - Static: 0.187 mW
  - Dynamic: 0.016 mW
  - Total: 0.203 mW

As regards the performance, the hardware module works in exactly the same way as the software module, where no error is committed. In the same way as software methodology, truncations are made to obtain integer values which represent the memory addresses where the desired pixels are located.

9.5.2.2 Histogram and equalization computation

The histogram process requires the image to be scanned to detect the number of pixels, each of these has 256 possible values, i.e. coding each pixel in one single byte. This process is performed several times, and is always followed by reorganization of the histogram values to spread the pixels over the complete possible range or to change the values of the pixels which are below or above a specified threshold thus simplifying the posterior processes. Considering all of these possible applications, two unique modules have been developed which can perform all the above mentioned functions: one to compute the histogram and a second to change the pixel values.

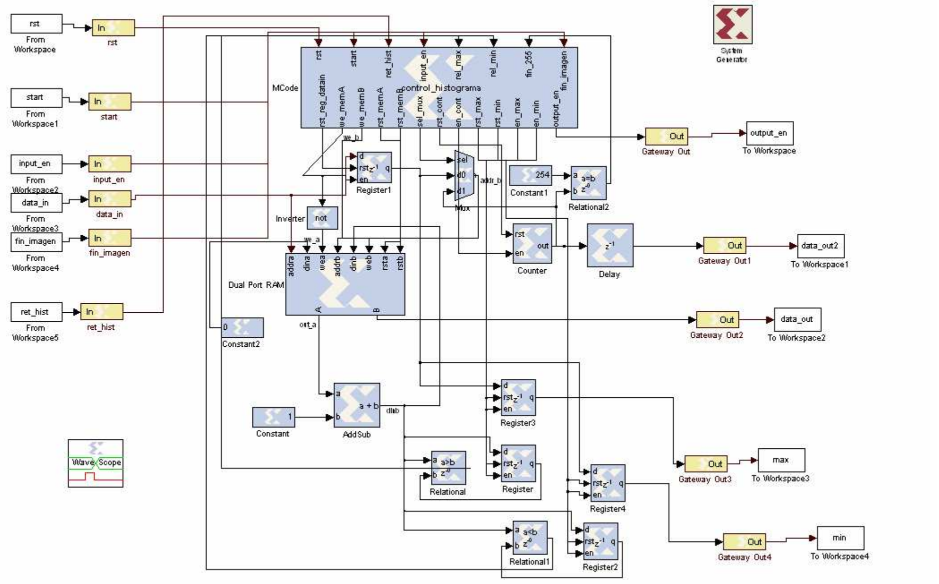


Figure 9.10: Hardware histogram module

As can be observed, the histogram is performed by a scanning process and stored in an internal memory of the FPGA. This memory can be implemented using two different kinds of FPGA blocks: Using distributed RAM blocks, which make the design faster or by using block RAM, which due to their physical location, use up more time when data is transmitted to them.

We have implemented a dual port RAM as it allows an address to be read at the same time as it writes in a separate address.With this solution, and by using a delay module and an accumulator, the histogram computation can be performed in as many cycles as the size of the image ($width \times length$) including a latency of just one cycle. Several other elements are used to read the resulting histogram such as delays, etc. These elements are used to read the memory and to consider the delays of the external memory.

The equalizer is a simple module which reads the values from the histogram and depending on two threshold parameters it stretches the histogram considering the values below one threshold as being black and those which are above the other parameter as being white. The implementation used is shown in Fig. 9.11.

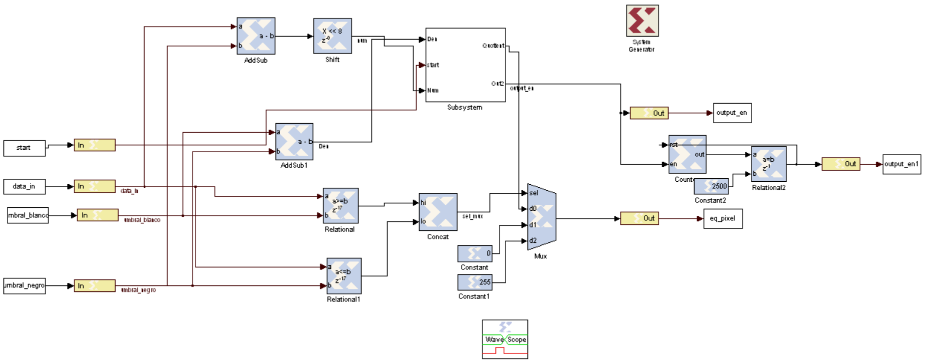


Figure 9.11: Hardware equalizer module

In this module, we have included the previously mentioned vision module. This division is used to perform the formula 9.2, used to stretch the values in the histogram space. When dividing by multiples of 2, as in the case of 256, the division module has been substituted by a shift register, which performs this operation using less hardware area.

$$Im_{int} = \frac{Max(Im_{ini}) - Im_{ini}}{Max(Im_{ini})} \tag{9.2}$$

$$Im_{str} = \frac{Max(Im_{int}) - Im_{int}}{Max(Im_{int})} \tag{9.3}$$

As has happened with the resize module, there are no differences between the hardware and software; in spite of this, we have used a division module which truncates the values into integers. There are no differences because, as has occurred before, the software truncates the values to integers, as pixels should take integer values from 0 to 255. The histogram hardware Implementation results are:

- Hardware area:
    - Look-Up table (LUTs): 144 (0.35%)
    - Flip-flops: 77 (0.17%)
    - Slices: 79 (0.39%)
- Time:
    - Frequency: 127.541 MHz
    - Total processing time for the resized image: 96.385 μs
- Power:
    - Static: 0.187 mW
    - Dynamic: 0.015 mW
    - Total: 0.202 mW

The equalization hardware implementation results are:

- Hardware area:
    - Look-Up table (LUTs): 52 (0.13%)
    - Flip-flops: 31 (0.07%)
    - Slices: 31 (0.15%)
    - BRAMs: 1 (2.5%)
- Time:
    - Frequency: 150.331 MHz
    - Total processing time: 81.759 μs
- Power:
    - Static: 0.187 mW
    - Dynamic: 0.014 mW
    - Total: 0.201 mW

The thresholds used to compute which pixels are considered as being white or black are image dependant. We have computed these values by considering that some images, due to illumination and other factors, can affect the pixel values and therefore, each image provides a threshold for white or black depending on the image examined. To compute these thresholds, the image histogram is scanned to search for pixels which are below or above a fixed value.

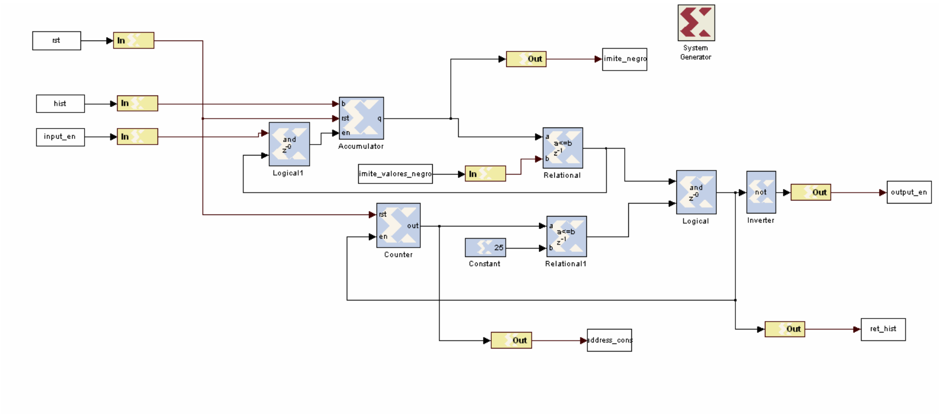


Figure 9.12: Black detection module

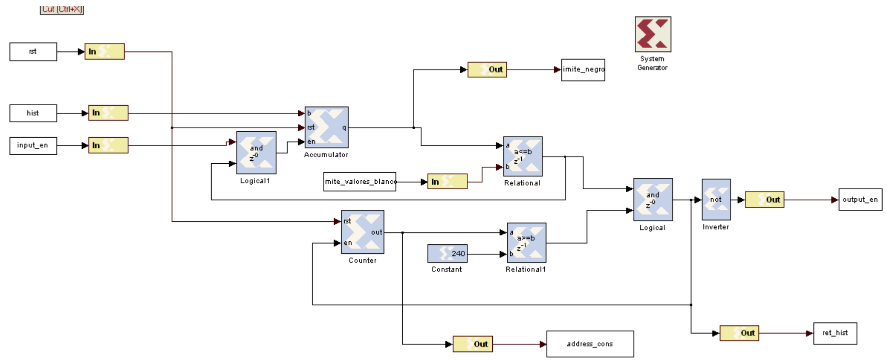


Figure 9.13: White value detection module

These values are computed dynamically by considering the pixels which are below a certain fixed value and below a value which can be modified whenever required. The modules used for this purpose are shown in Figs. 9.12 and 9.13. The results obtained for each module are the following:

- Hardware area:
  - Look-Up table (LUTs): 44 (0.11%)

  - Flip-flops: 18 (0.04%)
  - Slices: 24 (0.12%)

- Time:
  - Frequency: 173.853 MHz
  - Mean total processing time: 115.039 ns

- Power:
  - Static: 0.187 mW
  - Dynamic: 0.014 mW
  - Total: 0.201 mW

As regards the performance, it has been observed that there is loss in accuracy, this is because there is no truncation in any of the modules.

9.5.2.3 Finding the pupil

After resizing the image, and performing equalization based on the histogram followed by increasing the contrast to eliminate values below the threshold, the next step is to find the pupil within the image. The pupil search is performed in two steps:

- A coarse search: This is also divided into two separate processes. Searching for pupil values takes place by scanning the image in the horizontal direction and later in the vertical direction and vice versa, this is then followed by choosing the search which leads to a larger pupil radius.

- A fine search: Once we have detected the preliminary values for the centre of the pupil and its radius, a new scan is performed using the original image in only those zones which have been detected as possible boundaries from the previous step.

Coarse search

We start the iris search by scanning the image in both the horizontal and vertical directions to find the biggest block of pixels with black values. This can be observed in Fig. 9.14 where the coarse search for the pupil is performed in two main branches, these are different in the directions they scan the image, but identical in the functions they perform. For this reason and considering that both branches are memory data dependant, where it is not possible to perform them at the same time, the same hardware must be used for both.

The block used to scan the image is formed only by counters in the horizontal search case and by counters and multipliers for the vertical scan. However, once the desired pixels are detected, a search should be performed to find the largest dark pixel block, considering possible interferences provoked by flash reflections. This block is shown in Fig. 9.14.

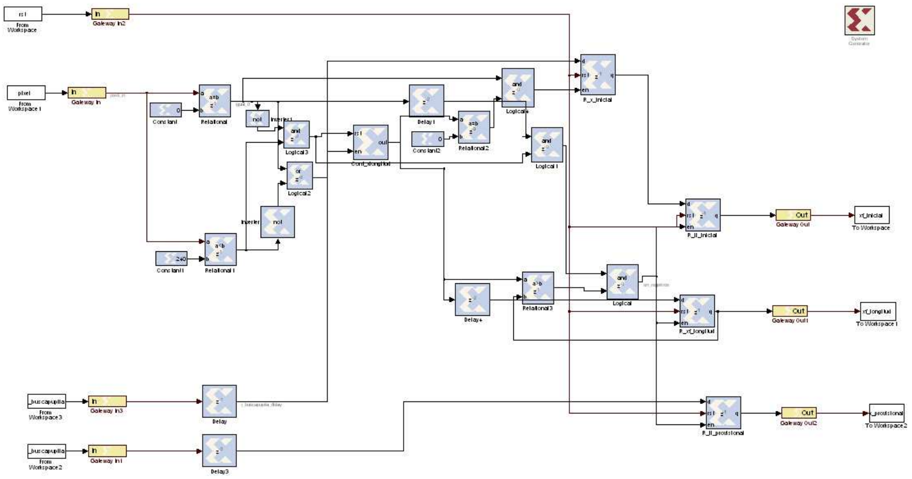


Figure 9.14: Hardware module to detect black blocks

This block is used to verify the image pixels and to store the initial value and the last value of the biggest block found. Additionally, it avoids pixels with bigger values than a preset threshold, as they are considered as flash reflections. For this purpose, the complete image is scanned and when a black pixel is found it is stored and the length of the line drawn from it in the horizontal or vertical direction is also stored, in case it is the biggest black area found.

Once the values have been located, the pupils coarse parameters are determined using the following blocks: one for the image scan in the vertical direction and a second for the horizontal case. After this first scan is completed, the systems radius is recalculated by means of a detailed examination of the diameter drawn in the opposite direction from the previous search. This process is performed by the block shown in Fig. 9.15.

The second branch is identical to this first one. Once both branches are computed, the values obtained are used to detect the coarse pupil values by using the block shown

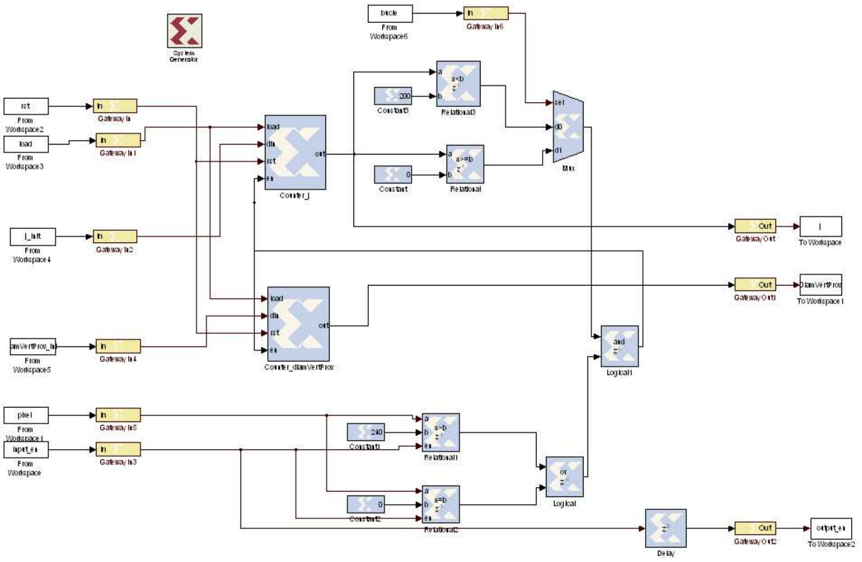


Figure 9.15: Hardware module used to detect coarse pupil values

in Fig. 9.16. In this block, a decision is made on which of the two diameters is bigger and therefore, a choice is made on the values from the corresponding branch.

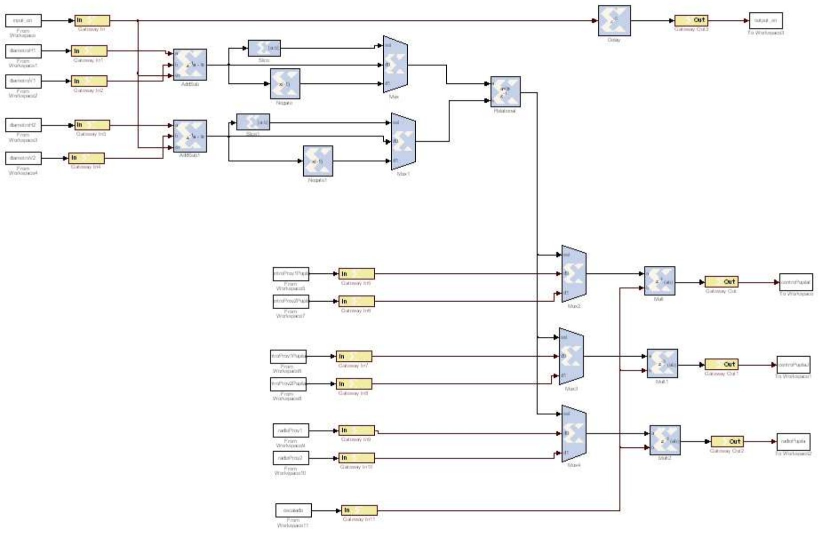


Figure 9.16: Hardware module used to detect final coarse pupil values

The complete pupil coarse search has led to the following results:

- Hardware area:
  - Look-Up table (LUTs): 2386 (5.83%)

  - Flip-flops: 1515 (3.70%)
  - Slices: 1267 (10.19%)

- Time:
  - Frequency: 60.536 MHz
  - Total processing time: 423.814 ns

- Power:
  - Static: 0.187 mW
  - Dynamic: 0.094 mW
  - Total: 0.281 mW

Fine search

The fine search is carried out by performing an image scan which considers different angles and different radii that surround the provisional pupil centre and that are beside the approximate radius. To implement this fine search, the module shown in Fig. 9.17 is used:

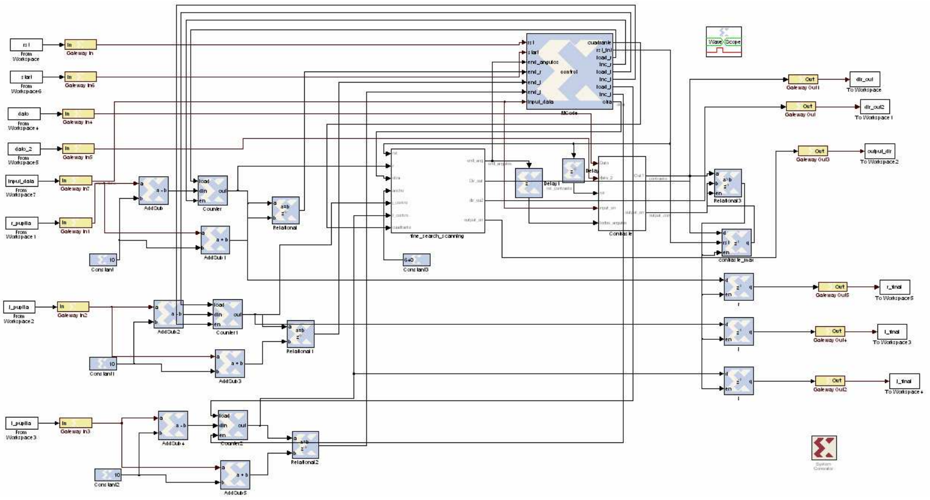


Figure 9.17: Hardware module used to detect fine pupil values

This module contains two sub-modules: the first is in charge of performing the fine search by considering the input values and the second is in charge of computing the difference in contrast between the pixels located in the same angle but with different radii. The first module controls the computation of the corresponding addresses required according to formula 9.4:

$$
\begin{aligned}
Diference = I(j + D_{j1}, i + D_{i1}) - I(j + D_{j2}, i + D_{i2})) \\
D_{i1} = r \times cos(\alpha) \\
D_{j1} = r \times sin(\alpha) \\
D_{i2} = (r - 1) \times cos(\alpha) \\
D_{j2} = (r - 1) \times sin(\alpha)
\end{aligned}
\tag{9.4}
$$

The second computes the difference in contrast between the pixel values obtained from the memory addresses.

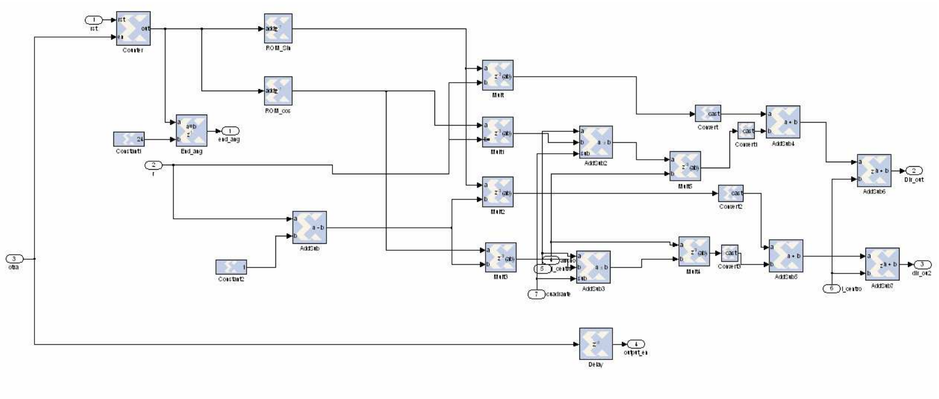


Figure 9.18: Hardware module used to search fine pupil values

The results obtained from this block are as follows:

- Hardware area:
  - Look-Up table (LUTs): 1171 (2.86%)
  - Flip-flops: 1015 (2.57%)
  - Slices: 687 (3.35%)
  - BRAMs: 2 (5%)
- Time:
  - Frequency: 81.646 MHz
  - Total processing time: 318.497 ns
- Power:
  - Static: 0.187 mW

- Dynamic: 0.094 mW
- Total: 0.281 mW

As regards the performance, errors have been committed in the address computations due to the truncations made in the sine and cosine angle values. However, the algorithm demonstrates how these errors are compensated when the maximum value is found.

9.5.2.4 Outer boundary

The outer boundary is again computed in two steps, first a coarse search followed by a fine search. The main difference between this search and previous boundary searches is that, in this case, both searches are performed according to the polar directions, i.e. examining different angles and different radii. The difference between the two steps of this module comes from the image examined: In the coarse search case, a reduced image is used, while for the second i.e. fine search, the initial image is scanned. The general module is shown in Fig. 9.19:

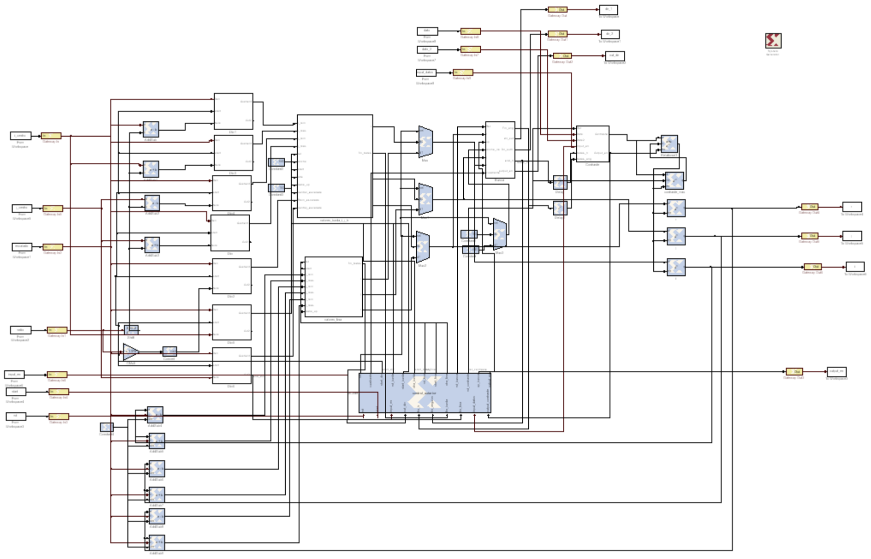


Figure 9.19: Hardware module used to detect the outter iris boundary

As the scanning is performed in both cases, following the same directions, two different modules have been implemented to compute the values of the horizontal and vertical directions for both searches. One of them operates on the coarse search and the other on the fine search. Both of them use the same module to address the memory, as this is common to both. The input parameters of this module are the pupil parameters previously found from the original image. In the first case, these values are divided by the scaling factor while for the second, the values required are obtained from the coarse search. The coarse search module is shown in Fig. 9.20:

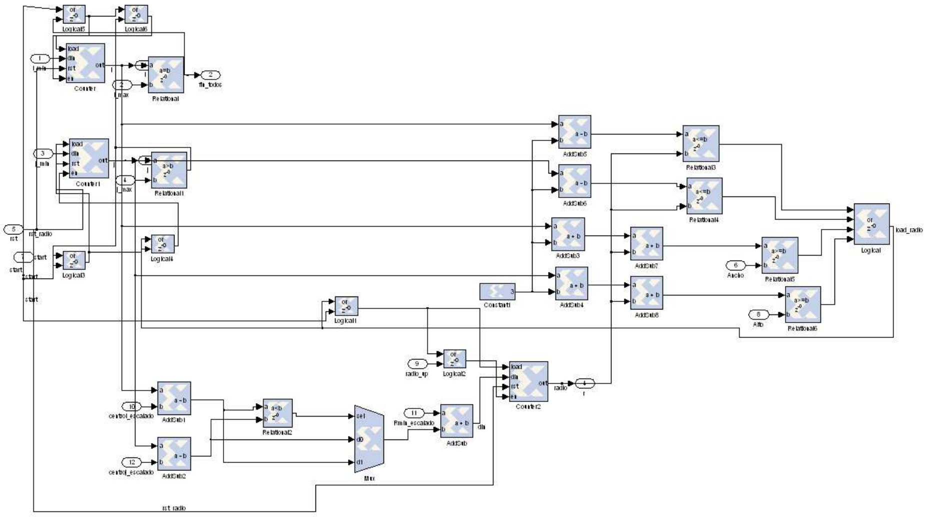


Figure 9.20: Hardware Coarse Search module used for the outer iris boundary

For the fine search module the scheme used is indicated in Fig. 9.21:

The general search block is shown in Fig. 9.22:

As in the inner boundary case, the contrast difference is again observed among the different image blocks. The results obtained for the iris boundary computation are:

- Hardware area:
  - Look-Up table (LUTs): 1985 (4.85%)
  - Flip-flops: 1404 (3.43%)
  - Slices: 1185 (5.79%)
  - BRAMs: 2 (5%)
- Time:
  - Frequency: 58.734 MHz
  - Total processing time: 408.622 ns
- Power:
  - Static: 0.187 mW
  - Dynamic: 0.031 mW
  - Total: 0.218 mW

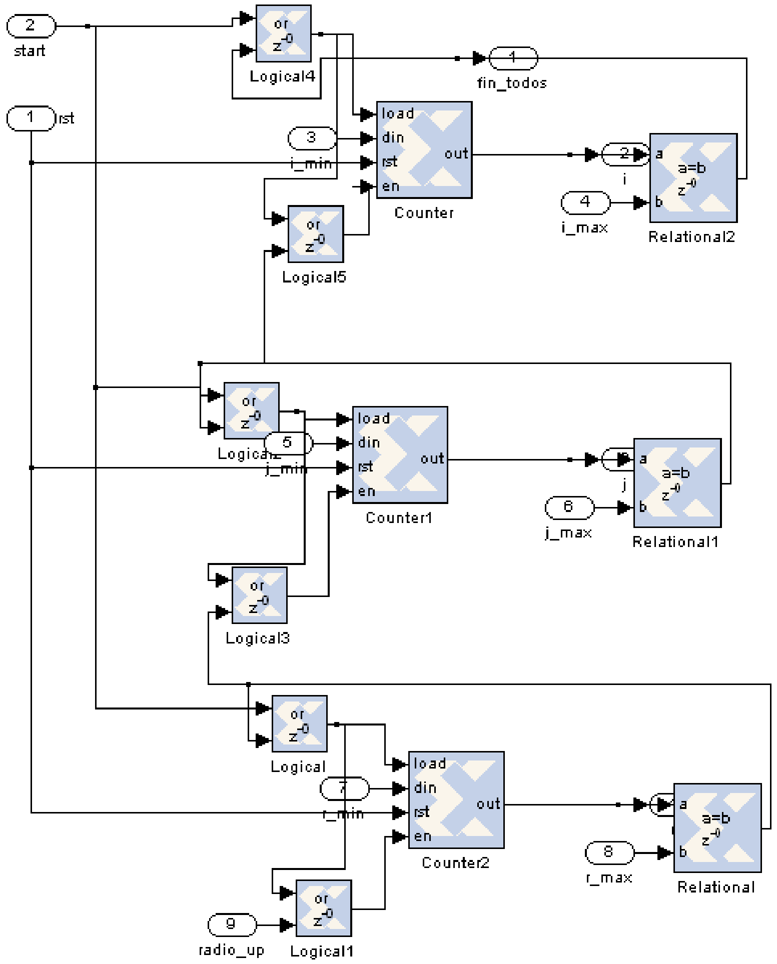


Figure 9.21: Hardware Fine Search module used for the outer iris boundary

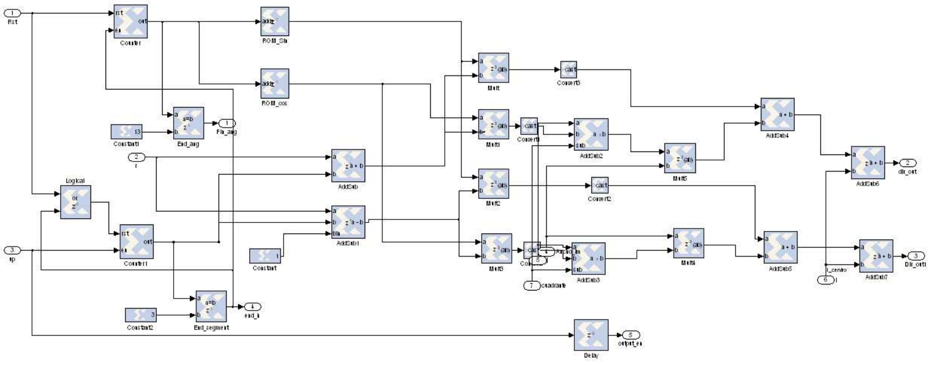


Figure 9.22: Hardware search module used for the outer iris boundary

9.5.3 Feature extraction

To perform the feature extraction two main processes have been considered: The iris signature computation and the wavelet transformation.

9.5.3.1 Iris signature computation

In the case of the iris signature computation, formula 3.15 indicates the requirement for the computation of trigonometric functions. Regarding hardware, there are two main methods commonly used to compute these types of functions: the use of a Look-up table or by using the CORDIC algorithm [64]. This algorithm relies on several shifts and iterations to compute the sine and cosine values. We have studied this algorithm and the Look-up table solution, both leading to similar results with respect to the hardware area. However, it has been decided to make use of the look-up table solution as the accuracy is less influenced by the algorithm and the response time is less when compared to the CORDIC solution which requires several clock cycles to return the result.

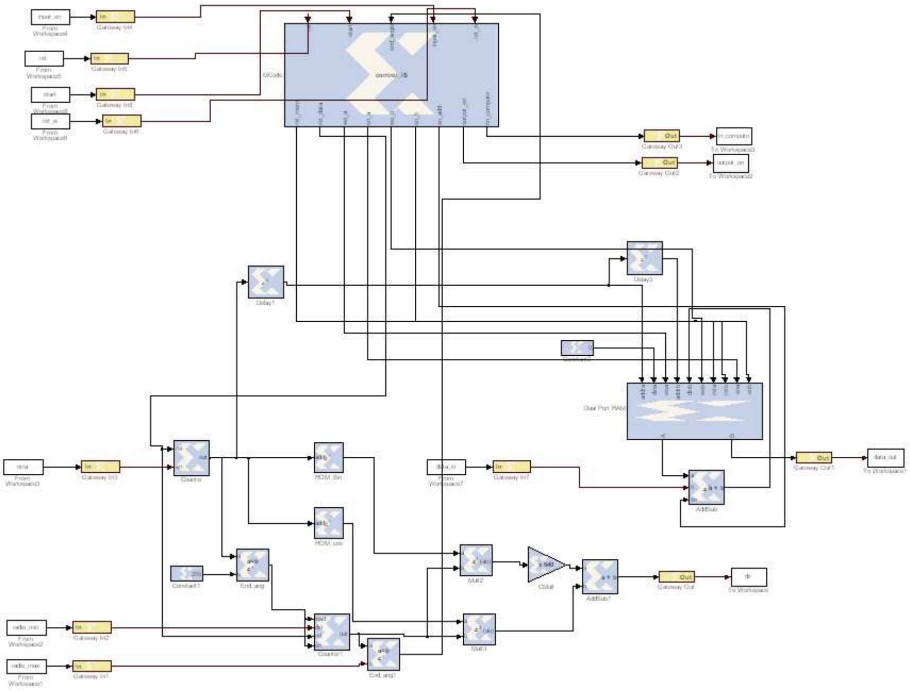


Figure 9.23: Hardware implementation of the iris signature computation

Considering the performance, both solutions introduce errors which are caused by the truncation of the number of bits used to represent the decimal numbers. Fig. 9.24 shows the differences obtained when computing the address of the Iris signature on a 400x400 pixel image. The values between -400 to 400 refer to errors that occur in the same row; however, values outside these boundaries report errors in different rows. These errors have been produced by the look-up table which has been used to compute the trigonometric functions for the iris signature computation. This value, due to lack of accuracy, has been observed to increase due to the posterior multiplication with other parameters. As Fig. 9.24 shows, these errors are mainly located in areas which surround the values of 350 and -100 and show an approximated symmetry of close to 150 on the abscissa. This symmetry and the error figure occur for two reasons. The first, is related to the accuracy of the LUT used for the trigonometric functions; this is because the output value of the LUT is set to 16 bits. The second is due to the later truncation performed on the iris signature computation after multiplying the LUT values with the pupil radius.

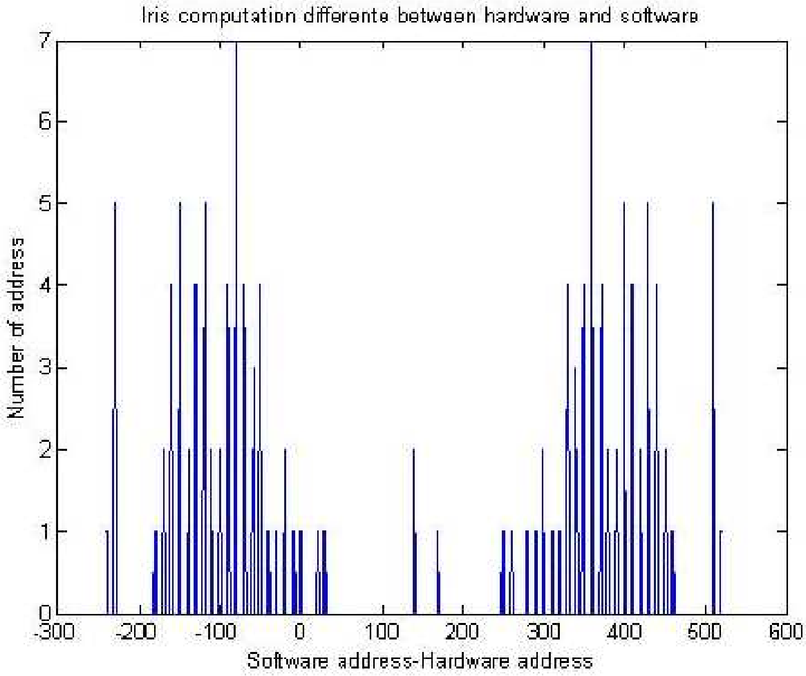


Figure 9.24: Differences between hardware and software iris signatures values

As regards the hardware results obtained for this module, the data obtained is:

- Hardware area:
  - Look-Up table (LUTs): 1845 (4.50%)
  - Flip-flops: 1325 (3.23%)
  - Slices: 935 (5.57%)

  - BRAMs: 2 (5%)

- Time:
  - Frequency: 90.234 MHz
  - Total processing time: 15.072 ns

- Power:
  - Static: 0.187 mW
  - Dynamic: 0.031 mW
  - Total: 0.218 mW

9.5.3.2 Wavelet transform

To implement the wavelet transform the approach proposed by Mallat in [93] has been followed. This approximation is called the fast wavelet transformation (FWT). It is based on the use of two filters (H and G) which model the wavelet transformation. After each filter is applied, down-sampling is performed to reduce the amount of data obtained and a new vector is generated which has the same size as the initial vector. After several experimental trials, it has been considered that the down-sampling of the vector can be obtained by inserting '0's in the following steps. To insert these '0's we have used FIFOs to implement and perform the corresponding shifts in these processes. The number of shifts is dependant on the wavelet scale considered; to control these processes, we have included control machines.

Fig. 9.27 shows the complete hardware implementation of the dyadic wavelet transformation. It may be observed that a FIFO has been included to control the input of the filters and therefore reuse hardware.

One of the major problems associated with filtering is in the multiplication process. The feature vector is formed by 8 scales of the wavelet transform, and thus, either 16 filters or 2 filters are required which are later reused. It has been decided to implement the second strategy as it requires less hardware area. By using this solution, once one scale is obtained, the low pass filter transmits the resulting vector to the H and G filters to perform again the filtering within the following time slot. When implementing this solution, the accumulative error due to truncations in the input vector should be considered, this is because it is not possible to develop hardware filters with unspecific input lengths. As regards the wavelet transformation, several tests have been performed according to the different data input lengths to the filters and by modifying the number

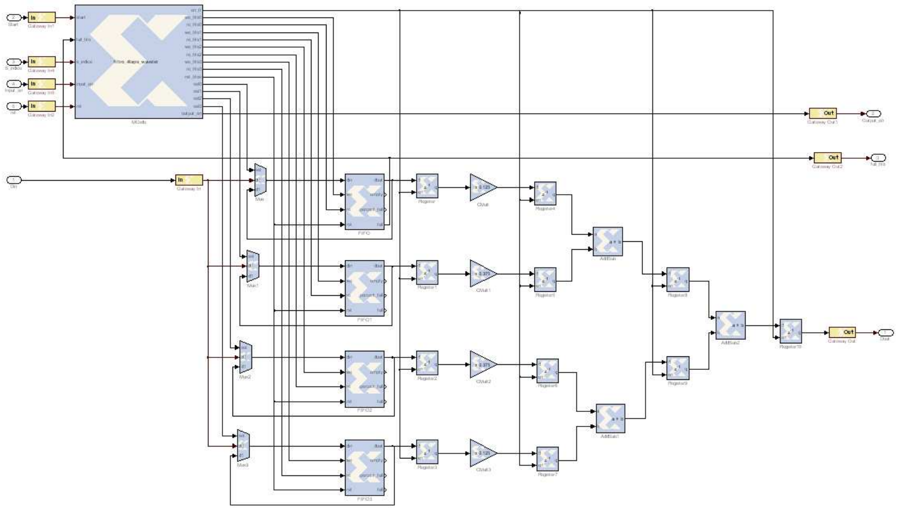


Figure 9.25: Hardware implementation of the FWT high band filter

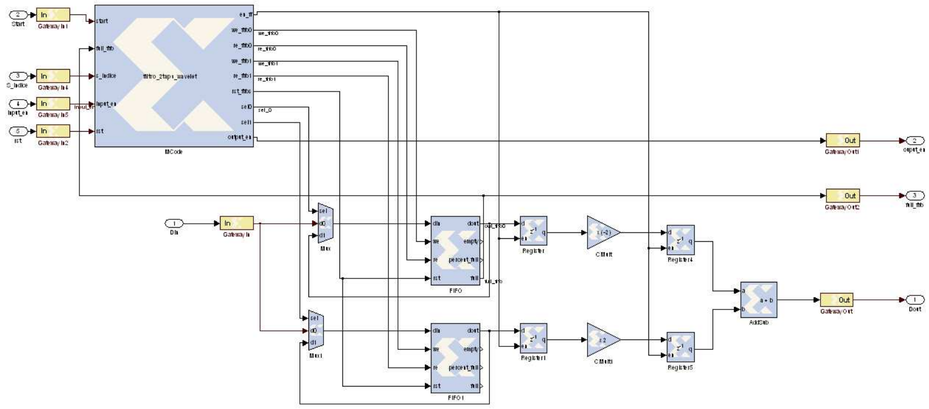


Figure 9.26: Hardware implementation of the FWT low band filter

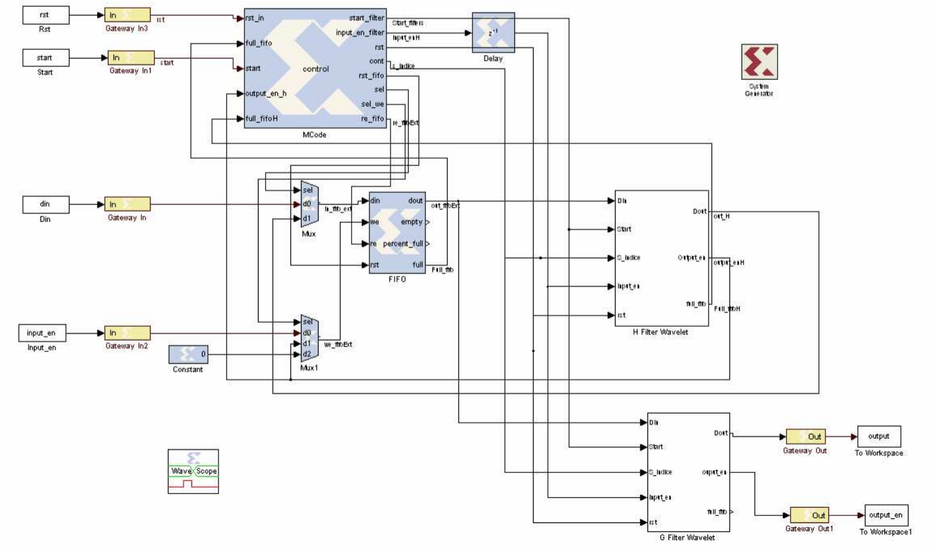


Figure 9.27: Hardware implementation of the FWT

of bits dedicated to the binary point values. We have considered three values: 10, 13 and 16 bits, where the integer part is set to 10 bits. A histogram which presents a comparative analysis between the software and hardware values obtained is presented in Fig. 9.28. It may be observed that when using integer arithmetic (red line, 10 bit width with no bits dedicated to the binary point) errors occur which range in value from -4 to 4, thus leading to errors in the feature computation. However, in the case where 13 or 16 bits are used, the amount of errors is reduced to less than one and the feature vectors obtained, due to the algorithm's robustness, are the same as those obtained from the software solution. In this figure it may also be observed that when using 13 or 16 bits no improvement in the results is appreciated; this is because the deviation from the results in both software and hardware is the same.

The hardware results obtained for the complete wavelet transform are:

- Hardware area:
  - Look-Up table (LUTs): 1259 (3.07%)
  - Flip-flops: 648 (1.58%)
  - Slices: 711 (3.47%)
  - BRAMs: 7 (17.5%)
- Time:
  - Frequency: 66.854 MHz

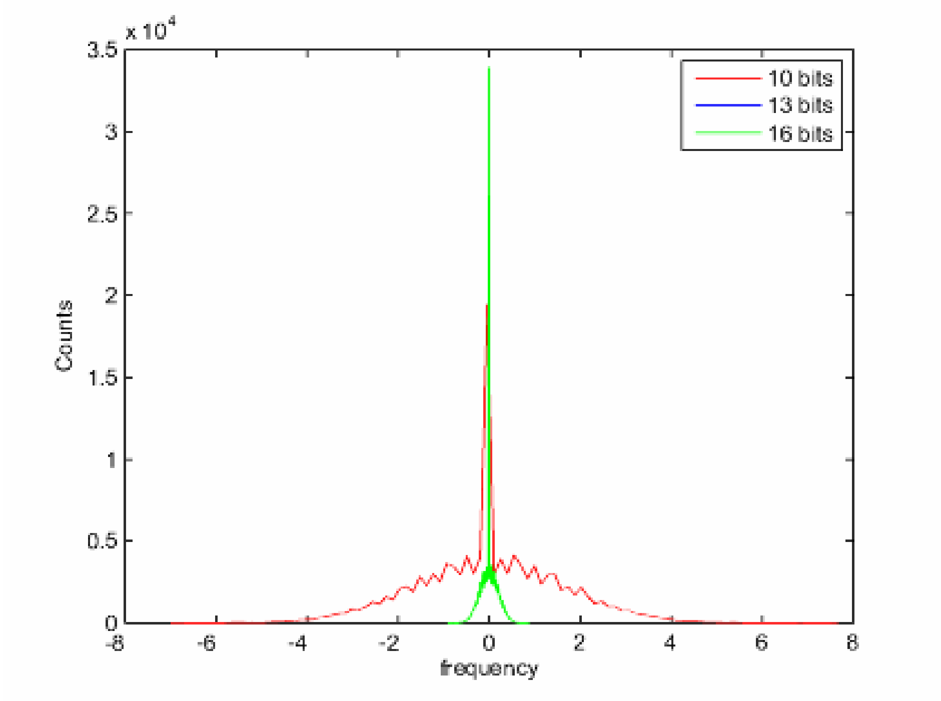


Figure 9.28: Differences between hardware and software FWT values

  - Total processing time: 64.0201 ns

- Power:
  - Static: 0.187 mW
  - Dynamic: 0.020 mW
  - Total: 0.207 mW

For the H filter alone:

- Hardware area:
  - Look-Up table (LUTs): 726 (1.77%)
  - Flip-flops: 407 (0.99%)
  - Slices: 726 (3.54%)
  - BRAMs: 4 (10%)
- Time:
  - Frequency: 87.604 MHz

    - Total processing time: 6.107 ns

- Power:
    - Static: 0.187 mW
    - Dynamic: 0.018 mW
    - Total: 0.205 mW

And for the G filter:

- Hardware area:
    - Look-Up table (LUTs): 332 (0.81%)
    - Flip-flops: 179 (0.44%)
    - Slices: 188 (0.91%)
    - BRAMs: 2 (5%)
- Time:
    - Frequency: 95.365 MHz
    - Total processing time: 5.610 ns
- Power:
    - Static: 0.187 mW
    - Dynamic: 0.016 mW
    - Total: 0.202 mW

9.5.4 Comparison module

The comparison process in the algorithm studied is the most straightforward part. It consists of checking the number of differences between the template and the feature vector obtained, i.e., the Hamming Distance. To implement this distance, several approaches may be carried out considering in all cases, the XOR between the two vectors, which will indicate which bits are different and which are not. To count the different resulting bits, five different implementations have been considered which are shown in Fig. 9.29.

It may be observed from Fig. 9.29 that different implementations of the Hamming Distance have been used, where all of these are performed in parallel using an XOR

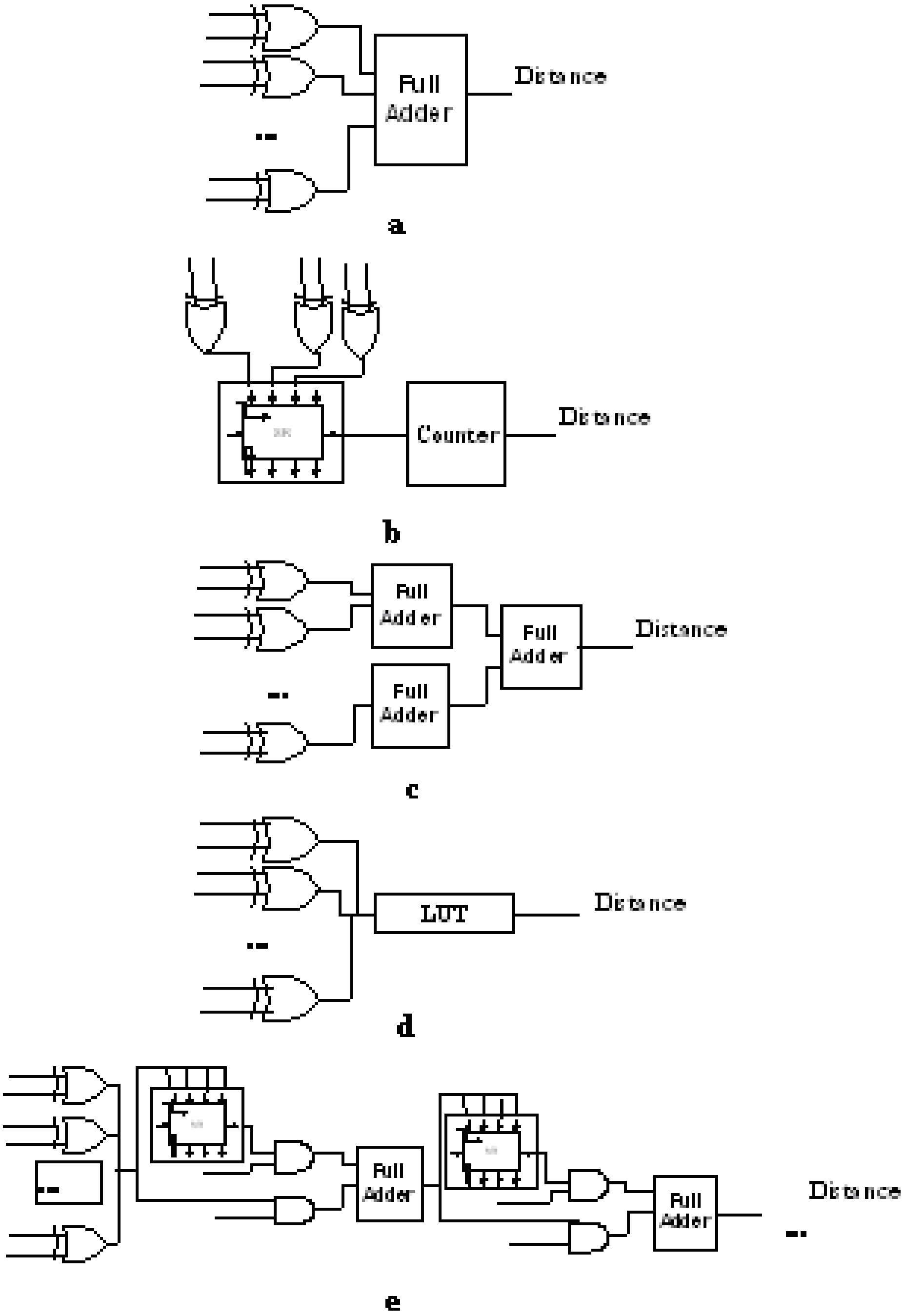


Figure 9.29: Different implementations of the Hamming Distance

operation on the template and code bits. The counting process for the number of '1's' contained in the resulting vector is carried out in different ways:

- using an adder,
- using a shift register with a counter,
- pipelined structure based on adders,
- a Look-up Table and
- a "magic numbers" implementation.

The first implementation relies on the mathematical formula itself. The input of the adder only considers the bits obtained from the XOR computation and the adder only adds them. This solution, although straightforward, requires an adder with as many inputs as the length of the resulting XOR vector. To reduce the number of inputs, the third proposal, which is based on a pipelined architecture performs this addition using several stages, and therefore reduces the area required by the adder. However, this solution is expected to be slower than the previous one as it requires several steps to compute the complete addition.

The second implementation is based on a counter. Here the value is increased when the enable counter signal is one. This counter is supplied with the resulting vector from the XOR after a shift register, which provides the counter with the aforementioned vector. Again, this solution, due to the shift register, is expected to be slower than the previous method as it must scan all the bits of the XOR vector one by one. The fourth proposal is based on a Look-up table. This table is addressed by the resulting XOR vector. This solution is commonly used in cases where the computational time is an important requirement or when the implementation of the operations is complex. The last proposal, called "magical numbers", proposed by Brian W. Kernighan and Dennis M. Ritchie in their book "The C Programming Language" ([75]) is based on a number of rotations and masks (Binary Magic Numbers) which are applied to each rotation. The number of iterations and masks depends on the vector length. All of these proposals do not lead to performance errors as no truncations or rounding up is performed during the computation of this value. As our algorithm, the Hamming Distance, is normalized using a fixed value, we have not considered this division so as to avoid truncations; and the comparison with the threshold has not been performed between 0 and 1 as is done in software but between 0 and 256*8 values.

Table 9.5: Hardware area results for comparison module

Implementation	Hardware area		
	Look-up tables	Flip Flops	Slices
XOR-Adder	125 (0.31%)	54 (0.13%)	73 (0.36%)
XOR-Shift Register - Counter	107 (0.26%)	108 (0.26%)	76 (0.37%)
XOR- Pipelined architecture	158 (0.39%)	63 (0.15%)	92 (0.45%)
XOR-LUT	6 (0.01%)	5 (0.01%)	4 (0.02%)
XOR-Binary Magic Numbers	212 (0.52%)	115 (0.28%)	127 (0.62%)

Table 9.6: Time results for comparison module

Implementation	Processing time	
	Frecuency (MHz)	$T_{wholevector}$ (us)
XOR-Adder	286.85	0.8925
XOR-Shift Register - Counter	270.27	9.472
XOR- Pipelined architecture	255.48	0.9786
XOR-LUT	260.38	0.984
XOR-Binary Magic Numbers	276.8	0.231

Table 9.7: Power results for comparison module

Implementation	Power		
	Static	Dynamic	Total
XOR-Adder	0.187	0.002	0.0189
XOR-Shift Register - Counter	0.187	0.002	0.0189
XOR- Pipelined architecture	0.187	0.003	0.019
XOR-LUT	0.187	0	0.0187
XOR-Binary Magic Numbers	0.187	0.004	0.0191

The results obtained for these implementations are shown in tables 9.5,9.6 and 9.7.

In this chapter, we have presented all the implementations of the different modules both on hardware and software. All the data obtained previously, permits us to perform the partition process that determines which process should be computed on each platform.

Chapter 10

Results obtained from iris ID tokens

In this chapter we will apply the partition method proposed and discuss different design alternatives of designing according to the requirements of the complete system. Before this, we will determine some of the parameters of the heuristic methods used to solve the partition problem.

10.1 Heuristic results

Once all the modules have been implemented using hardware and software, these values are considered as inputs for the algorithms required to obtain the optimum value of the cost function. To obtain the values of the parameters of the heuristic methods used for determining this optimum value the shortest processing time solution has been considered, where the configuration which leads to this solution occurs when all the modules present the best time results, i.e. their hardware implementation. In this section, the results obtained for the different heuristic methods used will be examined. These results are computed as the mean value of 20 computations for each method, where a random initial configuration has been considered.

In the case of the genetic algorithm, a population between N and 2N has been taken, where N is the number of genes in the chromosome as suggested in [3]. The maximum number of iterations has been set to 100.

In the simulated annealing, the temperature variation is computed following a geometric schedule, i.e. $Tnew = \alpha Told$ in the case where a better energy neighbour set is found, and $Tnew = \frac{T_{old}}{1-\gamma T_{old}}$ in the case where a penalty should be imposed.

Table 10.1: Processing time for different heuristic algorithms

Algorithm	Simulated annealing	Tabu search	Genetic algorithms
Time (s)	1.35	0.35	2.01

The value used for α is close to 1, where it is typically in the range from 0.9 to 0.99, and γ should have a low value to avoid getting caught up in an infinite loop. We have set the values 0.999 for α and 10^{-6} for γ. The starting temperature for these values is 400. Finally, the maximum number of iterations is 125 if the T_{stop} is not reached.

For the tabu search algorithm, the best results are obtained when the maximum number of iterations in the Tabu search is set to 100, where the maximum length of the Tabu list is 5 entries.

Table 10.1 shows the results for the time required to reach the optimal solution when using the different methods. Due to the random nature of the initial set, the results shown are obtained by taking the mean value of 20 different executions.

The mean search time as obtained from the Tabu search algorithm presents the best results when compared to all other methods. Genetic algorithms are efficient in obtaining solutions for these types of problems but it seems that for this specific case, based on a hardware/software solution, it does not present the best time results. Simulated annealing provides better results than genetic algorithms in terms of the search time, reaching the optimal partition in 1.35 s.

10.2 Architectures obtained

Now that we have studied the different implementations using both hardware and software, calibrated and presented the results which achieve the optimum solution for the cost function, we are now prepared to start combining them considering the proposal made for the ID token design. Several different approaches have been studied, considering different design factors: A secure ID token with a combination of minimal processing time, area, and power consumption. Our study will start with the proposal made for an ID token which achieves the minimum time, as this is the most common approach followed the majority of researchers.

For all of the previously mentioned approaches the dataflow analysis presented in section 3.5 is used, this considers the pre-processing block presented in Fig. 3.7 and the feature extraction block in Fig.3.10. As can be observed from these figures, node paths which operate simultaneously can be found, this is also the case for wavelet filters and others that may even occur at the same time due to the dataflow but where they cannot be performed concurrently due to the memory access. In these cases, where the nodes are external data dependant, we have considered, when possible, the process on the same platform so as to reuse hardware area. Fig 10.1 presents a summary of the dataflow algorithm. Following is a description of each module and their connections indicated in this figure:

- Modules 1, 2 and 3 are external data dependant from the memory, as they compute operations on the image, several on the initial image and others on the resized image, which is still large enough to be transmitted from one module to another. These should be computed consecutively, as the data required is that obtained from the previous module.

- Modules 4 and 5 can be divided into two sub-modules, indicated in the figure as 4.1, 4.2 and 5.1, 5.2, and refer to each of the search directions. These submodules are again dependant on data stored in the external memory. Due to this dependency and the sequential memory access, it is not possible to perform parallel computation.

- Module 6 is also memory data dependant and returns data to this memory.

- Module 7.1, 7.2, 8.1 and 8.2 are related to the search of the outer boundary. As in the case of the inner boundary search, these modules are data dependant and require memory to verify the pixel values. Therefore, these blocks cannot be computed simultaneously.

- Block 8 requires the data obtained from the pre-processing to compute the iris signature addresses as well as the image stored in the external memory to verify these addresses. This iris signature is formed from 256 values of 8 bits, and therefore, it can be transmitted from one module to another requiring 64 cycles to be transmitted, as the word length in the OPB bus is 32 bits.

- The wavelet transformation can be considered as two filters with a post downsampling (or zero-inserting). Input data of these two filters is the iris signature or the vector obtained from the previous scale iteration. This data dependency allows us to perform filtering concurrently.

- The zero-crossing and the matching process can operate at the same time as the wavelet transformation is being computed as Fig. 9.6 shows. These modules are data dependant on the previous block. The comparison module also requires access to the EEprom memory, but as it is the only block which accesses it, there is no problem in simultaneous access by any other module.

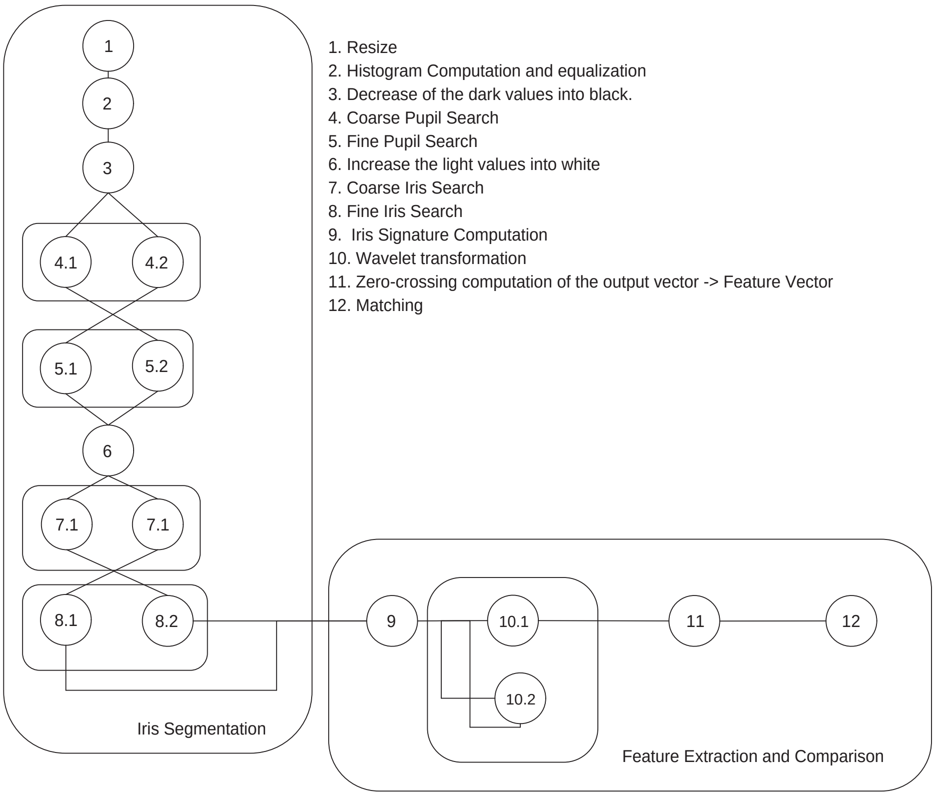

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.1: Iris algorithm flow chart

10.2.1 Minimizing processing time

When considering the data obtained from the previous implementations, we have to be aware that in a SoC solution just one clock is used for both the hardware and software, and therefore, the maximum frequency of the system is the lowest of all obtained. In our case, this frequency is delimited by the microblaze frequency: 40 MHz. Table 10.2 shows the processing time of the hardware modules considering this restriction:

From the table it can be seen that the processing time decreases when the values obtained are lower than those form the software solutions (column 3 shows the time saved as a percentage between the hardware and software solutions for each module).

If only the time is considered in the cost function, it is reduced to formula 10.1:

$$Cost = c_t * \sum_{i=0}^{n} Time_i \tag{10.1}$$

Table 10.2: HW, SW time results for each module

Module	SW Time	HW Time	%
Resize	0.0011	0.0003	70.93
Histogram	0.0008	0.0003	61.94
Equalizer	0.0005	0.0003	44.83
Contrast	0.0014	0.0009	32
Pupil Search	0.0009	0.0006	28.05
Iris Inner Boundary Search	0.0009	0.0006	27.77
Iris Outer Boundary Search	0.0006	0.0006	0
Iris Signature	0.0006	0.00003	94.10
Filter H	$8.92*10^{-5}$	$1.34*10^{-5}$	85
Filter G	$4.458*10^{-5}$	$1.337*10^{-5}$	70

In this formula, not only the modules have been considered but also the time losses in the transmission between the hardware and software modules for the communication required to send and receive the data from one platform to another. This time should only be computed when two adjacent modules are performed on different platforms and not between modules on the same platform. This time depends on the amount of data transmitted and the bus speed.

The resulting hardware/software architecture is that which reduces the majority of the cost function as the complete implementation is carried out in hardware, this is done for two reasons: It presents better time results in all the nodes and the time required for transmission between platforms has been avoided. Moreover, due to the absence of software modules, no data format conversion has been required in any of the performance processes, and therefore, no additional time is required for conversions. The partition obtained for this architecture is shown in Fig. 10.2. In this and future figures, modules performed using hardware are indicated in red and those performed using software in blue.

In this solution, we have been able to pipeline the feature extraction filters with the comparison, as the latency of the feature vector is relatively small in comparison to the time required to compute the matching comparison. Among the different possibilities for matching solutions, the method based on Kernighan's solution has been chosen, as this requires the least amount of computational time.

The complete processing time obtained is 8.12 ms and includes the deciphering of the initial image. This time makes it possible to use this architecture for high speed

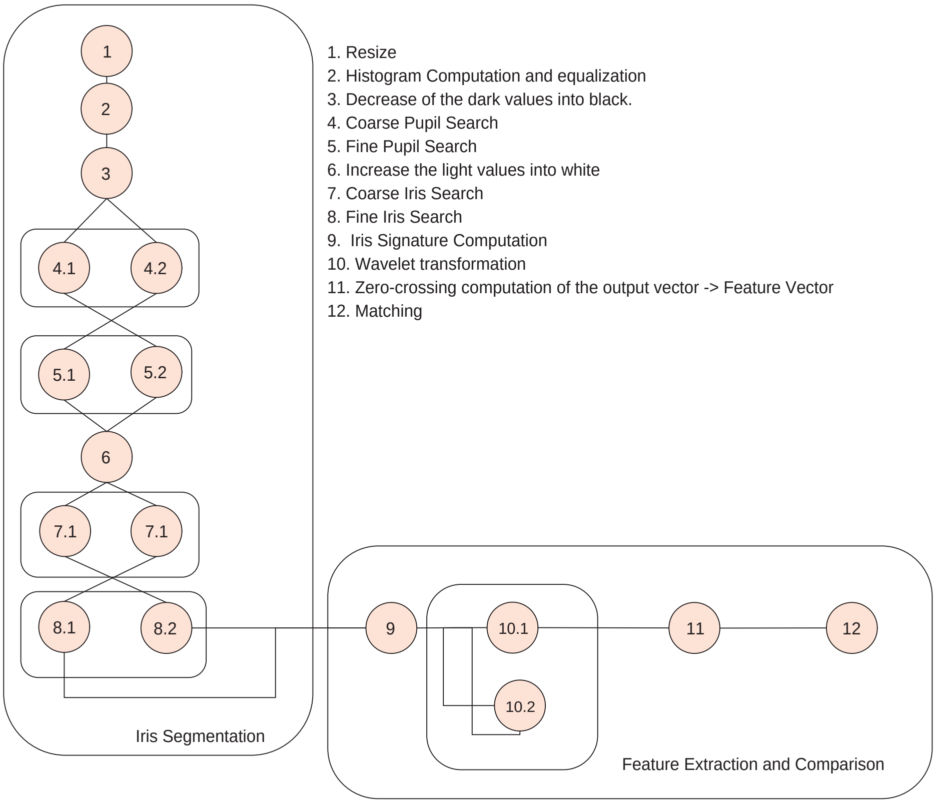

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.2: HW/SW partition for minimizing processing time

ID tokens which are required by systems where there is a high volume of simultaneous user accesses.

As regards other results obtained using this solution, the hardware area should be considered, where an important parameter associated which this is the complete occupancy of the token area. This area is split into several functions and is not only limited to the biometric algorithm. Within the same device, area is required for the microblaze processor and the AES encrypting/decrypting module. These modules are mandatory in all the approaches that have been considered, as the former is in charge of communications and the latter performs several vital tasks such as bus arbiter, control and interfacing with the outer world. Thus, it is important to satisfy the requirement that the complete area occupancy, including the area required for these two elements, is below the complete existing hardware area. The hardware area results obtained for the complete ID token are:

- Look-Up tables (LUTS): 32158 (78%)
- Flip-flops: 10252 (25%)
- Slices: 17980 (87%)
- BRAMs: 16 (40%)

As can be observed from this data, the requirements are satisfied, although the number of slices is clearly close to the maximum. If during the synthesis process, the area is considered as a main constraint, then this number can be reduced by using the Xilinx synthesize tool to minimize the hardware area as much as possible. However, in this case, priority has been given to time-reduction in the synthesizing process, as it is the key point of this approach.

The total power consumption of this solution is 0.677 mW. The static power consumption is fixed to 0.187 mW where this is due to the technology used and the dynamic power is 0.567 mW. In this approach and as a result of using a SoC solution, no static power is considered when the microprocessor and the dedicated hardware co-processors are not enabled, as they are all included on the same chip.

Regarding the performance, we have evaluated the complete ICE 2005 database with this approach where the results presented in fig 10.3 have been obtained.

This figure shows the differences mentioned between the data formats using hardware and software and the problems which arise due to truncation, as well as the influence on the systems performance. The above figure shows the complete FAR-FRR curve, the bottom curve only shows the behavior of these curves around the EER point.

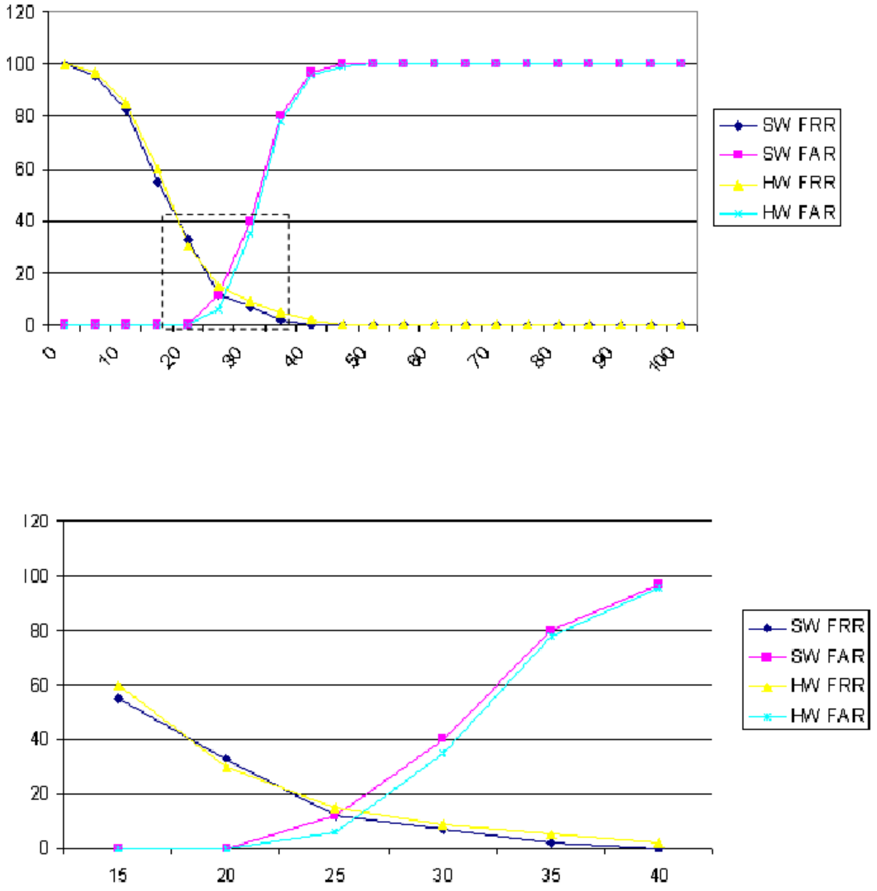


Figure 10.3: FAR vs FRR obtained in the time architecture

10.2.2 Hardware area and power

If the chief constraint in the design under investigation is the hardware area occupancy or minimum power consumed, the architecture obtained is the same for both of these cases, this is because the power consumed is directly related to the area occupied. The resulting configuration is shown in fig 10.4.

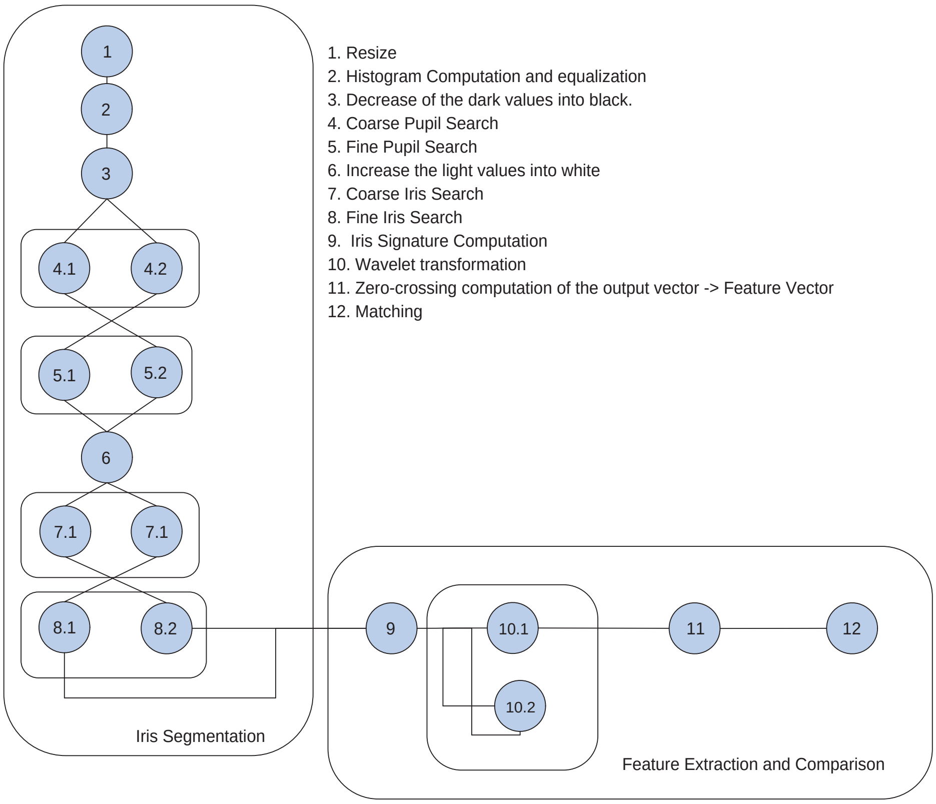

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.4: HW/SW partition for minimizing hardware area and power

The hardware area required in this solution is less than the previous approach, this is because the microblaze occupancy and the algorithm code do not require a high amount of slices and BRAMs. As a result of the reduced area, the power consumed by the system is lower than in any of the other cases. The hardware occupancy of this architecture is as follows:

- Look-up table: 19741 (48.19%)
- Flip-flops: 3160 (7.71%)
- Slices: 10737 (52.43%)
- Block RAMs: 8 (20%)

For this area and power, we should not only consider the occupancy of the microblaze and the code but also the AES module. The power consumption results are:

- Static Power: 0.187 mW.
- Dynamic Power: 0.043 mW.

Finally, since the encrypted image is stored within the token until the identification result is provided, the processing time required is 14,34696 ms. As can be observed, the time required is 30% more than the previous solution. This time increase is influenced by the absence of concurrence when using software solutions as opposed to hardware solutions. The performance results are the same as those shown in fig 3.11, as in this architecture all the data used is in floating point format, and therefore, no truncation errors are committed.

10.2.3 Performance

The previous solutions have been based on performing the complete biometric process on a single platform; however, the partition function can lead to other architectures which depend on the major constraint being considered. For the performance case, different solutions are provided, as when using dedicated hardware no errors were committed when the pre-processing was performed. It is the designer's decision to consider other constraints which depend on the systems requirements. For this reason, we have not only emphasized the performance term but we have also considered other terms such as time and hardware area. In this section, we will discuss some of these solutions:

10.2.3.1 Processing time

The first proposal presented could be the abovementioned for processing time reduction, where all modules are performed in the Microblaze. However, as we have already examined this possibility, we can move on to a further proposal (fig. 10.5) based on hardware and software. This proposal performs the pre-processing using the hardware and therefore, there is a significant time reduction without any loss of accuracy as the feature extraction is performed using software. The matching algorithm used is again the Kernighan's proposal as this solution is the most efficient for the reduction of the comparison time.

This solution decreases the computational time as the hardware implementations accelerate the pre-processing modules. The results obtained are:

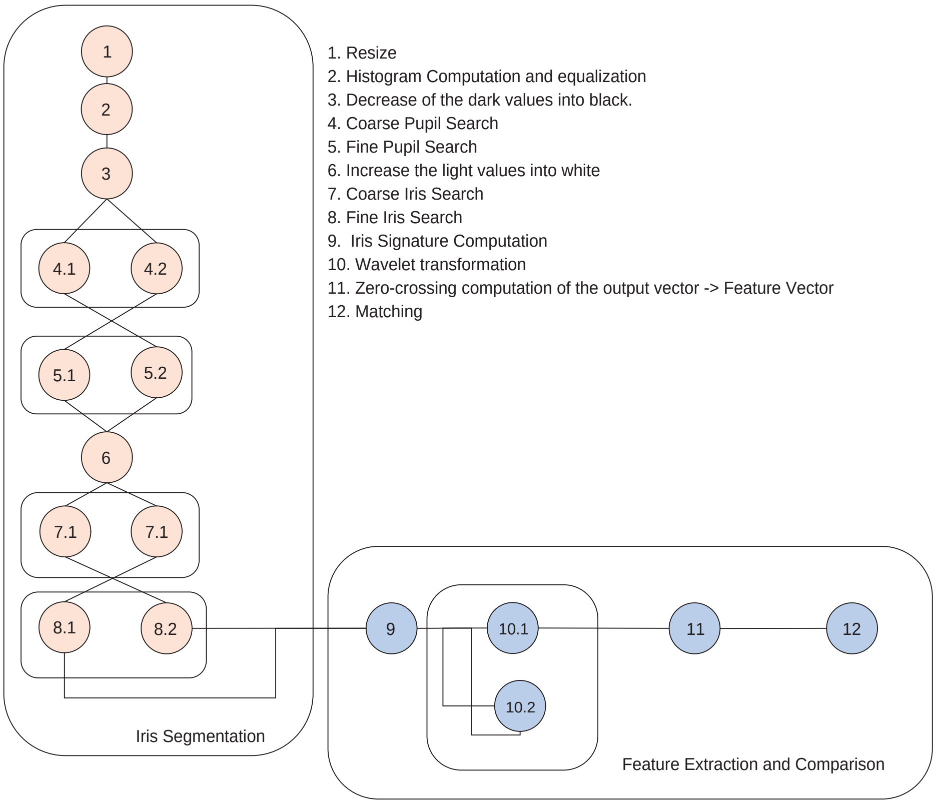

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching
9
10.1
10.2
11
12
Feature Extraction and Comparison

Figure 10.5: HW/SW partition to reduce the processing time while maintaining performance results

- Processing time: 10.234 ms
- Area:
  - LUTs: 25835 (60.07%)
  - FFs: 7471 (18.24%)
  - Slices: 14193 (69.30%)
  - BRAMs: 14 (35%)
- Power consumption:
  - Static: 0.187 mW
  - Dynamic: 0.137 mW

10.2.3.2 Hardware area and consumed power

To reduce the hardware area, we present this second approach (Fig. 10.6)

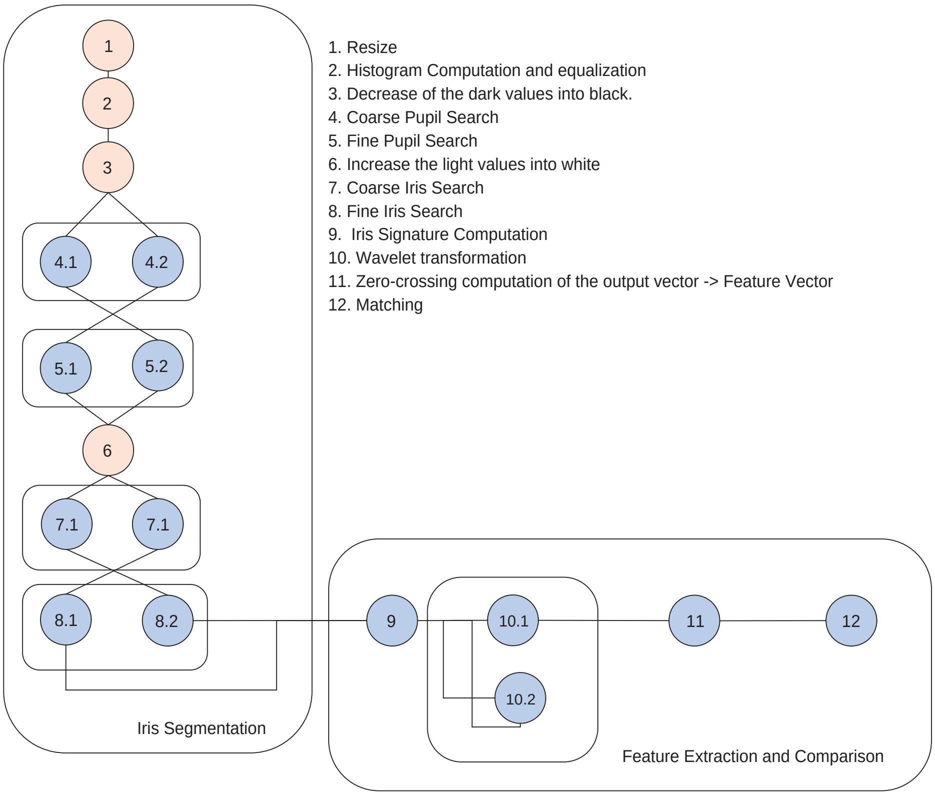

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.6: HW/SW partition to reduce the hardware area while maintaing performance results

To design this approach, we have slightly modified the values corresponding to the hardware area by introducing coefficient values in the partition formula of less than 0 for the pre-processing part (if these coefficients are 0 or greater the solution obtained attempts to reduce the hardware area and increase power optimization). The modules which are shown to have a negative coefficient are those where the percentage time reduction between the software and hardware is not greater than 10%. As the major constraint here has been to reduce the area, in this case, the comparison algorithm used is the XOR-LUT solution. The results obtained for this particular solution are as follows:

- Processing time: 10.354 ms
- Area:
  - LUTs: 20293 (49.54%)
  - FFs: 3501 (8.55%)
  - Slices: 11054 (53.97%)
  - BRAMs: 10 (25%)
- Power consumption:
  - Static: 0.187 mW
  - Dynamic: 0.043 mW

As can be observed, the complete hardware area is reduced due to the implementation of modules using software and only a slight increase in the processing time is observed.

10.2.4 Considering security Only

If an attempt is made to design a token where security is the main concern, the potential attacks that surround these systems should also be considered. In such a security based approach, we not only want to increase the systems security but also achieve an approach where the computational time is as low as possible. To do this, we have introduced further factors and considerations which demonstrate the benefits of our proposal. Here we examine the most vulnerable parts of the algorithm:

In Fig. 10.7 the data transmitted from one platform/module to another has been indicated. The data which requires an external memory to be transmitted due its size

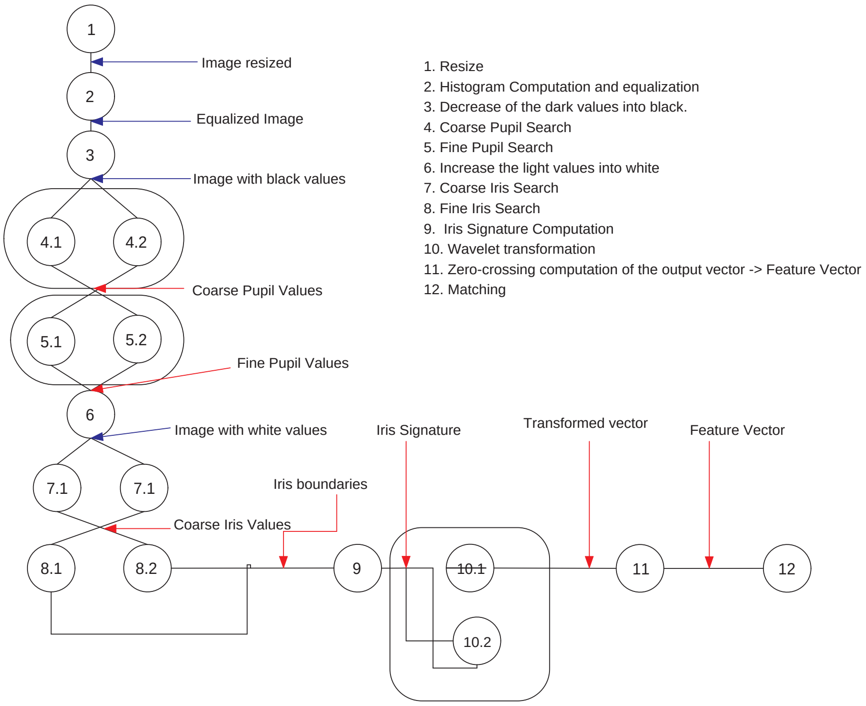


Figure 10.7: Vulnerable points in the iris algorithm flow chart

is shown in blue and in red the data which can be directly transmitted. This data is easily observed by any attack based on those presented in chapter 4, classified as authentication attacks. As has been pointed out, these attacks consist of listening to information for later replication of it or slight data modifications before applying it to the system. Among the different transmissions, certain types of data is more vulnerable than others due to the information managed; the closer to the end of the complete process, the more vulnerable the information is and the easier it is to replicate it for future system accesses. For this reason, both the feature vector and the matching vector require high levels of security. When determining which modules require more security, it is important to consider the algorithm, as in some cases, the modules are relatively simply permitting to recompose the algorithm and therefore, obtaining relevant data. This is the case for the zero-crossing representation of the wavelet transformation, which only consists of comparing if the resulting data from the wavelet is positive or negative. All these considerations have led us to the first proposal which is primarily based on security issues:

In this solution, it may be observed that the low pass filter of the wavelet, the zero-crossing and comparison process are performed using hardware, while the rest of the algorithm is performed using software. The wavelet filters can work simultaneously

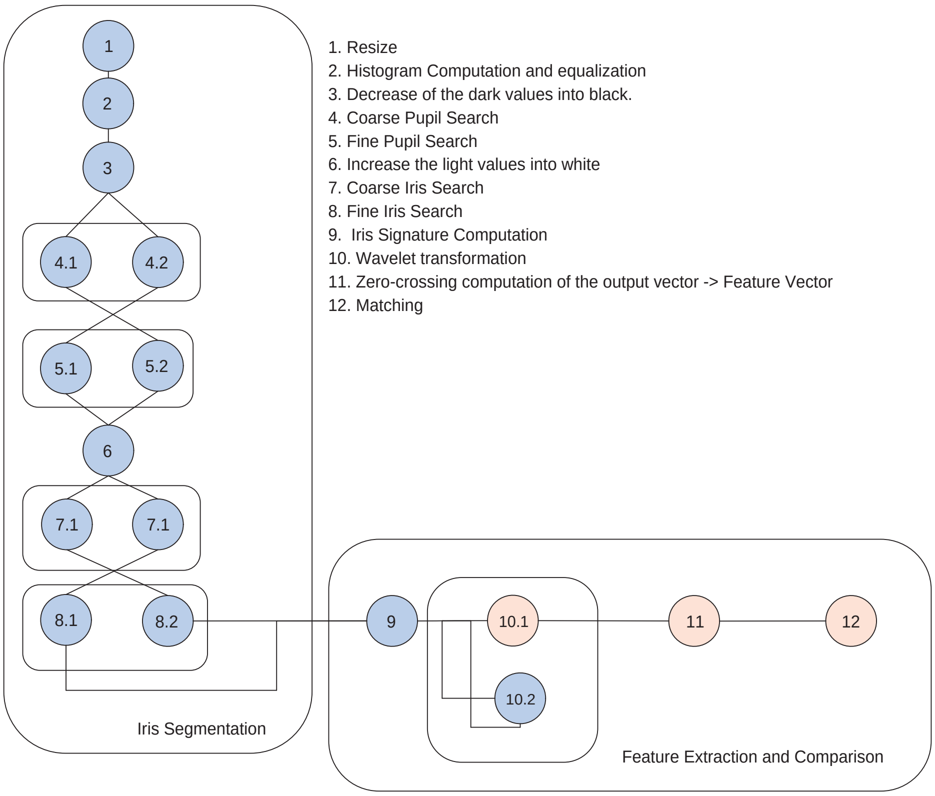

1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching
1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison

Figure 10.8: HW/SW partition to increase security

as both may be performed using hardware and software and where the transmission of the results to the low pass filter takes place once the high pass filter finishes. The zero-crossing and the matching work can also be carried out in parallel with these. For data transmission between the two filters, it has been necessary to include a floating point to fixed point conversion module where this increases the computational time. The rest of the algorithm has been implemented using software for hardware occupancy reasons.

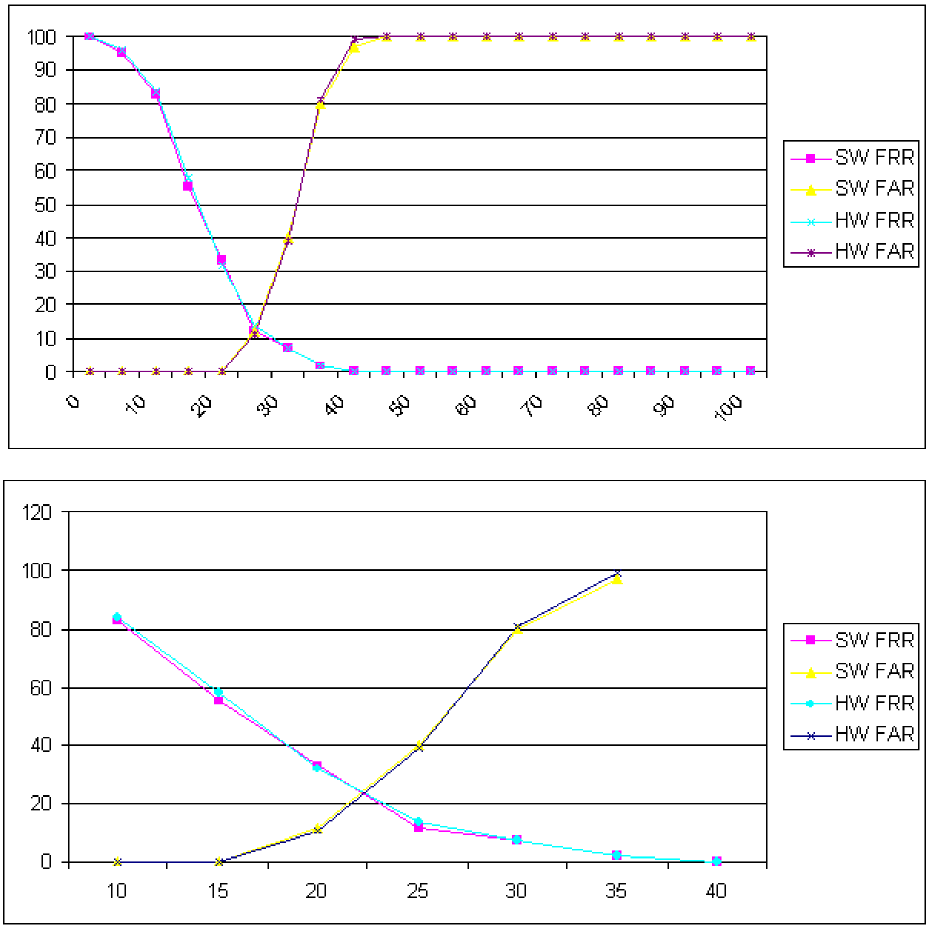


Figure 10.9: FAR vs FRR obtained in the security architecture

This solution also improves the performance results when compared to the all-hardware solution, as in the high band filter no accuracy is lost due to truncations. Therefore, no truncation errors are accumulated, leading to an implementation whereby part of the feature extraction is performed using hardware with no errors committed in comparison with the all-software solution. Small errors in the low band filters are almost entirely compensated by the zero-crossing module.

The results obtained for this configuration are:

- Processing time: 12.925 ms
- Area:
  - LUTs: 20323 (49.62%)
  - FFs: 3589 (8.76%)
  - Slices: 11175 (54.57%)
  - BRAMs: 9 (22.5%)
- Power consumption:
  - Static: 0.187 mW
  - Dynamic: 0.041 mW

Further examination of security issues shows that the transmission between modules should also be considered as they are an attack weakpoint. For this reason, modules which are data dependant, but do not require external memory, should be computed using hardware or software. This reasoning leads to the following configuration:

In this configuration, an attempt has been made to avoid all potential data transmissions between modules which occur at the same time. The processing time has been considered as the second priority factor for this design. As it can be observed, modules which require data from the external memory and which return data to the external memory have been performed using software. All modules which provide any data for transmission form one platform to another without the intervention of any external memory have been performed using hardware this is because such implementations provide higher security levels and reduce the computational time.

- Processing time: 11.883 ms
- Area:
  - LUTs: 28876 (70.49%)
  - FFs: 7963 (19.44%)
  - Slices: 14651 (71.54%)
  - BRAMs: 13 (32.5%)
- Power consumption:
  - Static: 0.187 mW
  - Dynamic: 0.15 mW

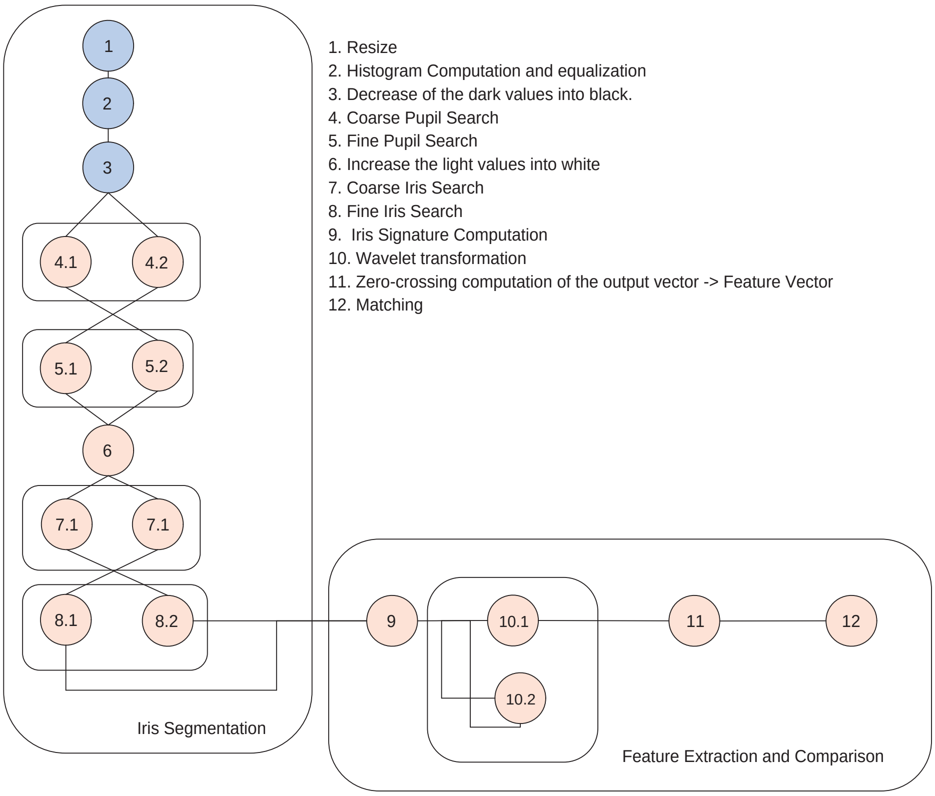

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.10: HW/SW partition to increase security avoiding the use of external memories

10.2.5 Combining all the factors

The final example of our proposal has been based on a system where several important constraints have been considered. To achieve this, use of the cost function is required. This new proposal attempts to effectively combine all of the previously mentioned factors and leads to a global solution. The solution obtained is shown in Fig. 10.11.

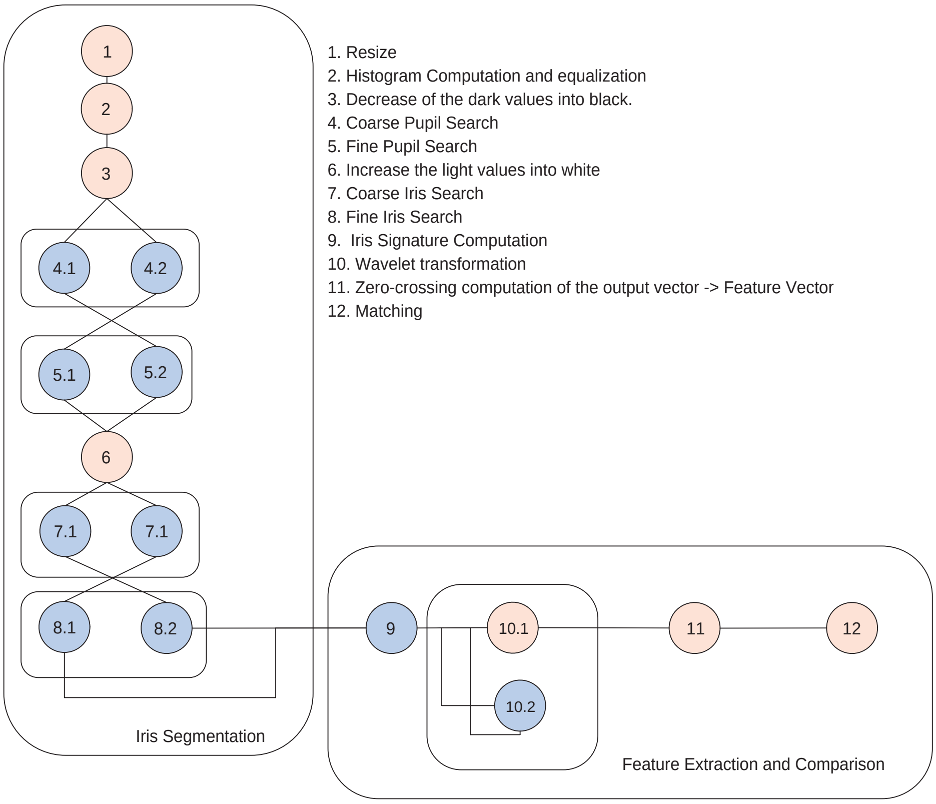

1
2
3
4.1
4.2
5.1
5.2
6
7.1
7.1
8.1
8.2
Iris Segmentation
9
10.1
10.2
11
12
Feature Extraction and Comparison
1. Resize
2. Histogram Computation and equalization
3. Decrease of the dark values into black.
4. Coarse Pupil Search
5. Fine Pupil Search
6. Increase the light values into white
7. Coarse Iris Search
8. Fine Iris Search
9. Iris Signature Computation
10. Wavelet transformation
11. Zero-crossing computation of the output vector -> Feature Vector
12. Matching

Figure 10.11: HW/SW partition where all factors have been considered equally

As can be observed, and by considering all the factors of the cost function equally, we arrive at a configuration where those modules which significantly decrease their processing time using hardware implementations are performed on such platforms, and those where the expected hardware area is significantly greater than software are performed using software, by reducing the space requirements the power consumption is also reduced. Additionally, the feature extraction presents simultaneous operation for the wavelet transform between the hardware and software, one for the low-pass filter computation and the other for the high-pass filter for security. These modules are followed by a zero-crossing representation and a comparison process performed using hardware, which decreases the chances of a successful intruder attack. The results obtained for this configuration are the following:

- Processing time: 8.597 ms
- Area:
  - LUTs: 20625 (50.35%)
  - FFs: 3680 (8.98%)
  - Slices: 11242 (54.89%)
  - BRAMs: 11 (27.5%)
- Power consumption:
  - Static: 0.187 mW
  - Dynamic: 0.085 mW

These results present an intermediate solution situated between the previous solutions proposed, where the processing time is slightly greater than the time obtained using the configuration which has presented the minimum time. The area is also larger than that obtained for the all-software configuration, but significantly lower than the minimum-time solution. Regarding security, it is important to indicate that this solution is more secure than the other solutions, however less secure than the previous solution presented where security has been of maximum priority.

10.3 Conclusions

Throughout the last three chapters, a proposal for a biometric system design based on ID tokens has been presented. We have offered several indications on how this system should be designed by considering all the different factors such as communications security and the functionality of the different elements. The research performed has been focused on the development of an ID token device where this design has been based on a hardware/software co-design. This particular design attempts to obtain the different advantages associated with each type of platform used. For this purpose, a partition function has been proposed, with a scheduling algorithm which follows the principles exposed for the complete system. The proposal made has been evaluated using an iris ID token, where we have developed all the modules using both software and hardware platforms and by later combining these in accordance with the different constraints, such as area, time or security. Several architectures have been obtained

Table 10.3: HW/SW Resume of the different configuration obtained

	Time	Area/Power	Performance		Security		Combination
			Time	Area	Time	Area	
Area (Slices)	17980	10737	14193	11054	14651	11175	11242
Time (s)	8.12	14.35	10.23	10.35	11.88	12.93	8.95
Dynamic Power (mW)	0.19	0.043	0.137	0.06	0.15	0.06	0.08
ΔEER	1.4%	0%	0%	0%	1.4%	0.5%	0.5%

showing the feasibility of our proposal as well as an analysis on how different aspects influence the final results.

Table 10.3 presents a summary of the architectures obtained. As can be observed, the processing time, the area and the power consumption obtained vary according to the design criteria. In the case where it has been attempted to reduce the processing time, the time obtained is less than that obtained for the rest of the configurations, however, the area is seen to increase significantly, this is because all of the biometric processes are performed using hardware. Other criteria considered, apart from the processing time, are those where the area and power consumption are to be optimized, here all the biometric processes are performed using the microprocessor, and therefore, both the area and power consumption are reduced significantly. When attempting to obtain better solutions where priority has been given to the performance or security, two possibilities have been examined: the first considers the processing time, and the second considers the area. The solutions demonstrate how an attempt is made to satisfy the second requirement, however, the solution does not present the same results as given by previous configurations, this is because the first requirements should be satisfied. Finally, a configuration where all of the above mentioned requirements have been considered with equal priority has been presented. This configuration demonstrates a balance between all of them, showing the possibility of having more than one requirement satisfied on the same ID token.

Chapter 11

Iris Biometrics search engine

The most common applications of hardware/software co-designs in Biometrics are related to time reduction for the recognition process. As previously mentioned, most authors' efforts have been dedicated to developing co-processors which compute biometric tasks, however these have a high computational cost [89], [49], [48], [47]. These solutions gain a significant reduction in the computational load of the main processor and accelerate the complete systems processing. In this chapter, we will follow this approach and study how co-designs can help minimize the systems processing time required for computation. We will show how it is not only necessary to perform the majority of the high-time cost functions using hardware, but also the importance of using optimized distribution and access for the different system elements. For this purpose, we have developed a biometric search engine, a system where the time requirements are critical and the computational load provoked by millions of comparisons is relatively high. We will describe two architectures based on hardware/software co-designs: the first one is based on a central architecture where all the elements are connected via the central processor, and the second, where memories are directly accessible by the module which most frequently uses them. The first approach will show how transferring recurrent information provokes a significant waste in time, while the second design demonstrates a time reduction when compared to the first approach.

This chapter is organized as follows: The motivation behind this work is described in the following section, where the necessity for a search engine by current societies is pointed out and how this is covered by an iris recognizer. This is followed by a depth analysis of the matching algorithm implemented and its characteristics which make it suitable for the current application. Our first attempt is presented in section 11.3. Thanks to a more in-depth study of this preliminary trial, we have verified several constraints that provoke wasted time, this has lead us to our final proposal.

The final proposal is described in section 11.4, first by introducing the modifications to the system to solve the aforementioned problems and finally by detailing the complete architecture. Results from both architectures are presented in section 11.5, where a comparative analysis with software solutions is presented. This is followed by details of future work in this area concerning the integration of larger databases for both architectures. Finally, the conclusions of this work are presented.

11.1 Motivation

Everyday, globally, many situations require some type of identification, several of these applications as a mandatory task and others to improve general performance. In most cases, the number of possible users to be detected is relatively low, e.g. restricted access to company information or physical access to a building; however, in other situations, the identification application is required for millions of people, this is commonly referred to as massive identification. Such identification processes require accurate methods to clearly differentiate one person from another as identities should be found among millions of users. These systems are commonly used in several scenarios such as large company access and border controls, where the time for user recognition should be kept to a minimum.

Border controls in many countries have to deal with the problem of illegal immigration, i.e. people from poorer or conflictive countries entering through the back-door of richer countries, in search of improved living conditions. The number of illegal immigrants increases everyday. Many of them may never reach the desired land on their first attempt, they are intercepted by border patrols and repatriated. Many of these rejected immigrants make several attempts to falsify border controls, by not providing any identification so as not to be identified in following attempts. border control offices have to deal with this recognition problem for people who do not want to be recognized, and thus, they check for a positive identification using information stored in databases which contain the identification of those who have already attempted to enter the country.

Massive identification is also a problem for human welfare organizations who carried out food distribution in poor countries to cover basic necessities. Much of this aid does not reach the population as it is managed by mafias. In this situation, correct identification of those who receive food can minimize the cause of incorrect distribution and improve aid provided to those who really need it.

But probably the best known situation where massive identification is required is related to proscription lists. These lists are used by many companies to reject a client if previous activities are not accepted by the company or by other organizations, i.e. in the case of denying hooligans access to sporting activities. Many proscription lists can be found nowadays, from hooligans to debtors lists, and of course, with varying extensions. Checking one or several lists can take days in some cases; and in many situations this time is not available or convenient.

In all such cases, the service provider should recognize the user he/she is dealing with among many users in a relatively short time span. Therefore, if an illegal immigrant tries to access a country on a second occasion or on subsequent attempts, the border control office can identify him/her for immediate repatriation; the Human Welfare Organizations could check and identify those who receive their aid; or sport organizations could sell season tickets only to those supporters who are known not to be violent.

Although the usage of Biometrics is increasing day by day and several modalities are being studied, just two of these offer commercial search engines: Fingerprint and face recognition [79], [12]. When considering iris recognition, the main problem for search engines with respect to massive identification is the impossibility of performing a pre-classification of the sample to be measured [123]. Thus, as opposed to other modalities, there is no way of reducing the search to a limited number of users. Each time a search is performed, all the users stored in the system have to be verified, this increases considerably the computational time. Even with the most straightforward matching algorithms known, Iris Biometrics require an extensive amount of time for massive identification applications. Therefore a new solution must be found.

11.2 Iris biometric matching

An iris search engine basically works as an identification biometric system 3.4. It is formed of different processes which lead to a feature vector that is then compared with those previously stored in the database and determines whether the sample being studied belongs to one of the users in the database. This matching algorithm, in the search engine, increases significantly the complete computational time as it should be performed several times. This drawback has motivated the work presented in this Thesis, i.e. biometric matching. When compared to the rest of the processes, the matching procedure causes the highest computational load on the complete system.

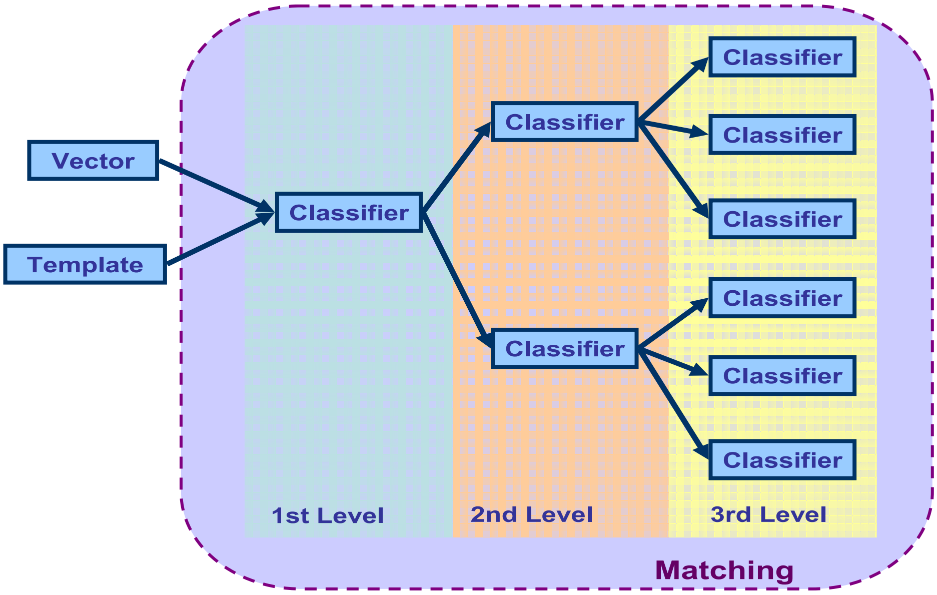


Figure 11.1: Classifiers in a tree structure

In most modalities [123], matching follows a tree structure formed by different classifiers which perform comparisons in several steps. First, the feature vector is classified into any of the first level branches; each branch is also divided into smaller branches, to simplify the classification process. However, even though some authors have studied the possibility of making an iris pre-classification, the best results obtained are only close to 95% [119] for correct matching, this is insufficient for massive identification scenarios. As mentioned in 3.4, several matching algorithms have been proposed in the literature 3.4, however, our work has been based on the algorithm presented in [25], which returns a feature vector, named iriscode, of 256 bytes. The algorithm chosen provides outstanding results as the feature extraction algorithms should provide a large interclass distance and a small intra-class distance, this makes it suitable for massive identification applications. The matching algorithm used is a modification of the Hamming distance, shown in [25].

$$HD_{norm}(A,B) = 0.5 - (0.5 - HD_{raw})\sqrt{\frac{n}{911}} \tag{11.1}$$

$$HD_{raw}(A,B) = \frac{\|(Code_{A\_Rotated} \otimes Code_B) \cap Mask_{A\_Rotated} \cap Mask_B\|}{n} \tag{11.2}$$

$$n = \|Mask_{A\_Rotated} \cap Mask_B\| \tag{11.3}$$

where $Code_{A\_rotated}$ and $Code_B$ are the template and sample vector. Norm is defined as the number of bits equal to '1',and $\otimes$ represents the exclusive OR (XOR) function, $Mask_{A\_Rotated}$ and $Mask_B$ are masks obtained during the pre-processing and the feature extraction block, by locating which bits of the feature vector should be considered for measuring differences and which bits come from iris zones occluded by eyelashes or eyelids, so as not to consider these bits in the comparison. The 911 factor is an empirical factor obtained as the mean value of the bits measured and $\cap$ represents a logic AND operation. This formula will be employed several times on the same template and sample, but for different rotations of the template and its associated masks. These rotations cater for possible head tilts or movements during the positioning of the iris in the initial image. As a result of the feature extraction block [28] [25], rotations should not be performed only on one bit, but performed every 16 bits, as each point considered leads to 16 bits of the vector and not just a single bit. This formula provides a distribution similar to the one shown in Fig 11.2. This figure was obtained by Daugman and published in [25], and shows the comparison between 625000 iriscodes from different users. It may be observed that the comparison of different users leads to Hamming distance values in the order of 0.5. However, when a user is compared with his/her template, all the bits tend to be in agreement, so the resulting Hamming distance is nearly 0. As a result of the low standard deviation of the distribution when comparing different users, it has been observed that this algorithm is suitable for massive identification systems.

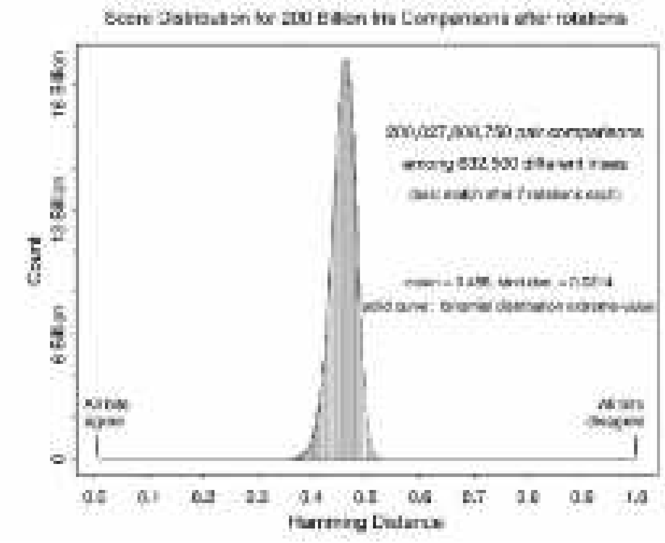


Figure 11.2: Score distribution for 200 billion comparisons [25]

11.3 Initial architecture proposal

The initial approach to build a search engine for iris recognition is based on conventional centralized architectures. This type of architecture is characterized by a unique central microprocessor which controls all the peripherals required for the task to be performed. Surrounding this unique microprocessor several peripherals and co-processors can be found, including those for developing specific tasks. We have verified the limitations of our proposal. These are provoked by a central architecture scheme, which will be discussed in the following subsections. However, this first attempt and its study has helped to present a new and more consistent proposal which reduces all such problems and leads to a more efficient architecture. Therefore, to better understand the final proposal, it has been considered of interest to introduce this first approach and its analysis as a means to reduce time consumption in other applications. As has been previously mentioned, the objectives laid out for the search engine have been to perform massive comparisons for reduced processing times, i.e. computing the matching algorithm several times. Although equation 11.2 does not indicate many possibilities for parallelization, it is clear that dedicated hardware is faster than software, thus the matching procedure should be implemented using hardware, leaving other processes to be performed by the microprocessor, such as those referred to as control and dataflow.

11.3.1 Matching

The modified Hamming distance formula is formed by several operations, some of which are easily implemented in hardware, while others are not. For future explanations, we consider the numerator of the equation to be:

$$Numerator = \|(Code_{A\_Rotated} \otimes Code_B) \cap Mask_{A\_Rotated} \cap Mask_B\| \tag{11.4}$$

And the denominator as:

$$Denominator = \|Mask_{A\_Rotated} \cap Mask_B\| \tag{11.5}$$

Therefore, the Hamming distance can now be expressed as:

$$HD_{norm} = 0.5 - \left(0.5 - \frac{Numerator}{Denominator}\right)\sqrt{\frac{Denominator}{911}} \tag{11.6}$$

As can be seen from this equation, after these values have been computed, two divisions must be computed (*Denominator/911*, *Numerator/Denominator*) and a square root; finally, this is followed by further standard multiplications and subtractions. The implementation of these operations requires additional hardware area and increased time, as such implementations are generally carried out by iteration algorithms [82], [116], [62]. With respect to the computing time expected and truncation provoked by the algorithm, we have verified a separate solution: a Look-Up table. This Look-Up table stores all the possible Hamming distance values that may be retrieved, and its addressing is determined by the numerator and the denominator obtained each time. Using this solution, the accuracy problem is reduced and the time to obtain data is accelerated. Both the numerator and denominator are seen to represent a count of the bits which are equal in both vectors.

Computing equality in hardware is translated into an XOR operation, meanwhile the bit counting can be implemented in different ways, as has been explained in the previous chapter. A comparison of these implementations can be found in [88] [87]. It has been indicated in these papers that each implementation is suitable for different applications which have a variety of requirements. For this special case, where hardware area is not relevant but time is a critical parameter, the best implementation is based on a unique full adder.

With regards to the iriscode rotations, these can be performed by the hardware or the microprocessor. In the first approach, rotations were performed by the microprocessor. Rotations are 16 bits long and the memory word length is 32 bits as well as the input length of the matching co-processor, thus, the microprocessor should not only access the RAM memory, but also consider that in some cases a rotation will require two reading accesses and a later concatenation before transmitting the vector to the co-processor.

11.3.2 Central processor

The microprocessor in this type of architecture is mainly used to perform sequential processes and for communication with peripherals, using dedicated hardware for these purposes as it does not contribute to time reduction. The main microprocessor functions are:

System control: As a central processor, it should control which processes are running and according to the system performance which processes should be initialized, etc. Additionally, it should check possible errors reported from different modules and solve them or inform the user.

Master of all peripherals Numerous peripherals are connected to the central microprocessor for several purposes: input/output communication, storage, etc. The communication between them and the processor should be initiated by the central processor and indicate the state of each one of them.

Dataflow: Data is transformed by the norm co-processor; however, the dataflow control is carried out by the central control unit, i.e. asking for data, when necessary, from the RAM memory and transmitting it to the processors, etc.

User Interfacing: Since the central processor is the master input/output peripheral, each time data is received from the outer world the microprocessor is in charge of interpreting this data to detect the user's commands or to store it in the correct location, it is also in charge of returning the results obtained.

11.3.3 Peripherals

As previously mentioned, several peripherals are included in the platform for storage and communication purposes. These peripherals and the selection criteria followed are detailed in this section:

User interface: The chosen user interface is a serial interface, such as the RS-232. This protocol is proposed as it is a relatively simple protocol and most devices avail of this port. Other possibilities, with increased speed characteristics, may also be used, e.g. USB connections and parallel interfaces. When interfacing, a user can introduce the data required for comparison, but this can also be carried out automatically by connecting the biometric system to this interface machine, i.e. the system proposed here can work as a peripheral of a major system.

Ethernet connection The decision to use an Ethernet connection to download the database has been based on time and feasibility reasons. First, an Ethernet connection is one of the fastest connections; and second, the database server/computer is most likely to have at least one of these ports. This connection is used as a dedicated port, i.e., it does not follow any UDP or TCP/IP protocol, so maximum speeds of up to 100 Mbps are available as no overheads are added.

SRAM Memory The RAM memory is used to store the database templates and their corresponding masks. The choice for using this type of memory has been determined by the access time, as these memories are faster than non-volatile memories. Its size (number of addresses) is fixed by the database size, as this data is stored in it. All the memories studied address 32 bit long data. As the vector length used is 256 bytes, and 32 bits are read in each access, 64 accesses are required for the complete template, i.e. 128 accesses for the complete template and its associated mask.

Non-volatile memory The non-volatile memory is used to store the Hamming distance values, this removes the need to perform square root and division operations presented in formula 11.6. The size of this memory depends on the range of the numerator and denominator, as the memory addresses are computed using these two values; if the range is [0, 2047], the size will then be 4 MB addresses. On the other hand, the system accuracy is determined by this memory word length and the range of values to be represented. In massive identification systems, accuracy becomes a major concern as two different users may lead to a Hamming distance which can be confused with a distance considered for the same user (e.g. a word length of 8 bits only allows a representation of 256 (28) values. If we consider a uniform distribution of the values represented by them, the number 0.349 is represented by the same binary number as 0.353, by setting the threshold to 0.35, the system will display an error in this situation)

11.3.4 Architecture proposal

This approach has been the first step to designing a complete search engine. The block diagram can be seen in Fig. 11.3. This search engine is composed of several peripherals, dedicated hardware for matching purposes, and a microprocessor that controls the dataflow.

When initializing the system, one should first download the database (assumed to be located in a computer or server) to the SRAM Memory. Once the database is stored in the system, it waits until a user-interrupt occurs. This interrupt is provoked by the RS-232 protocol, and provides the feature vector to be verified, the mask associated with it and if desired, a threshold used for comparison.

After receiving the feature vector, the comparison process starts. Matching is done with the help of the dedicated hardware, as was previously mentioned.

The microprocessor reads all the stored templates from the RAM memory one by one, along with their corresponding masks. Several accesses should be made to complete the whole template. When the templates and masks are completely read, the distance formula is applied by employing the dedicated hardware.

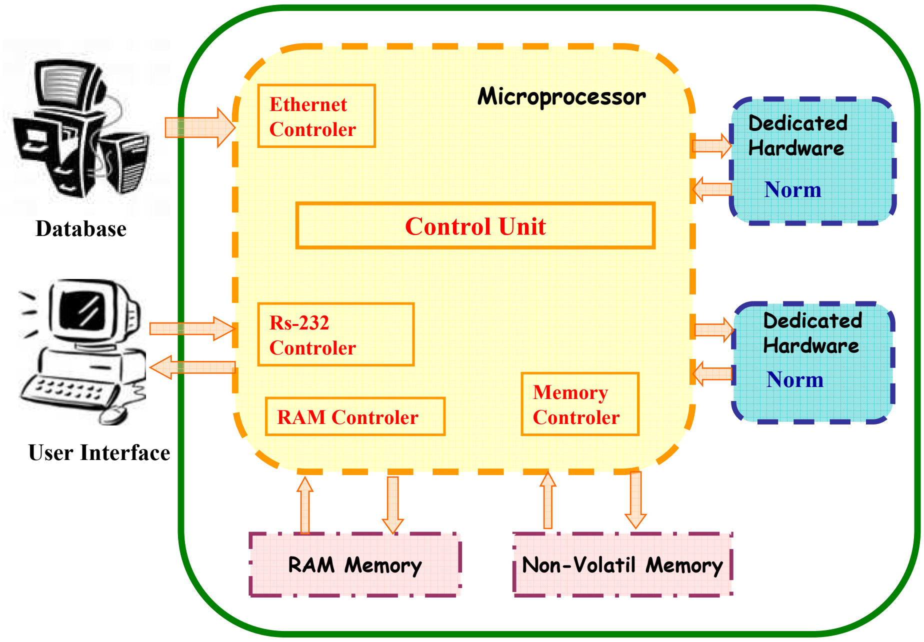


Figure 11.3: First search engine approach

As explained, this is done by accessing the LUT stored in the flash memory, this solution reduces the hardware complexity and computational time.

The distance obtained is compared to the most similar distance which was obtained previously. If it is lower than this distance, the new distance is saved as the best distance and also the user associated with it.

Afterwards, the template and its mask are rotated and the matching block is again called to compute the distance between these two vectors. Once all rotations have been computed, another template is read from the SRAM Memory.

The processes finish when the system detects that the complete RAM memory has been read; the system then returns the user associated with the best distance found and the distance via the serial interface. If a threshold was initially provided, the resulting distance is compared to it. If the distance obtained is below it, it is considered to belong to the user found. However, if it is not, the sample studied is considered not to belong to any of the users in the database. If no threshold has been provided, the distance obtained is compared to 0.255 to determine if it corresponds to the user's template from the database being verified [25].

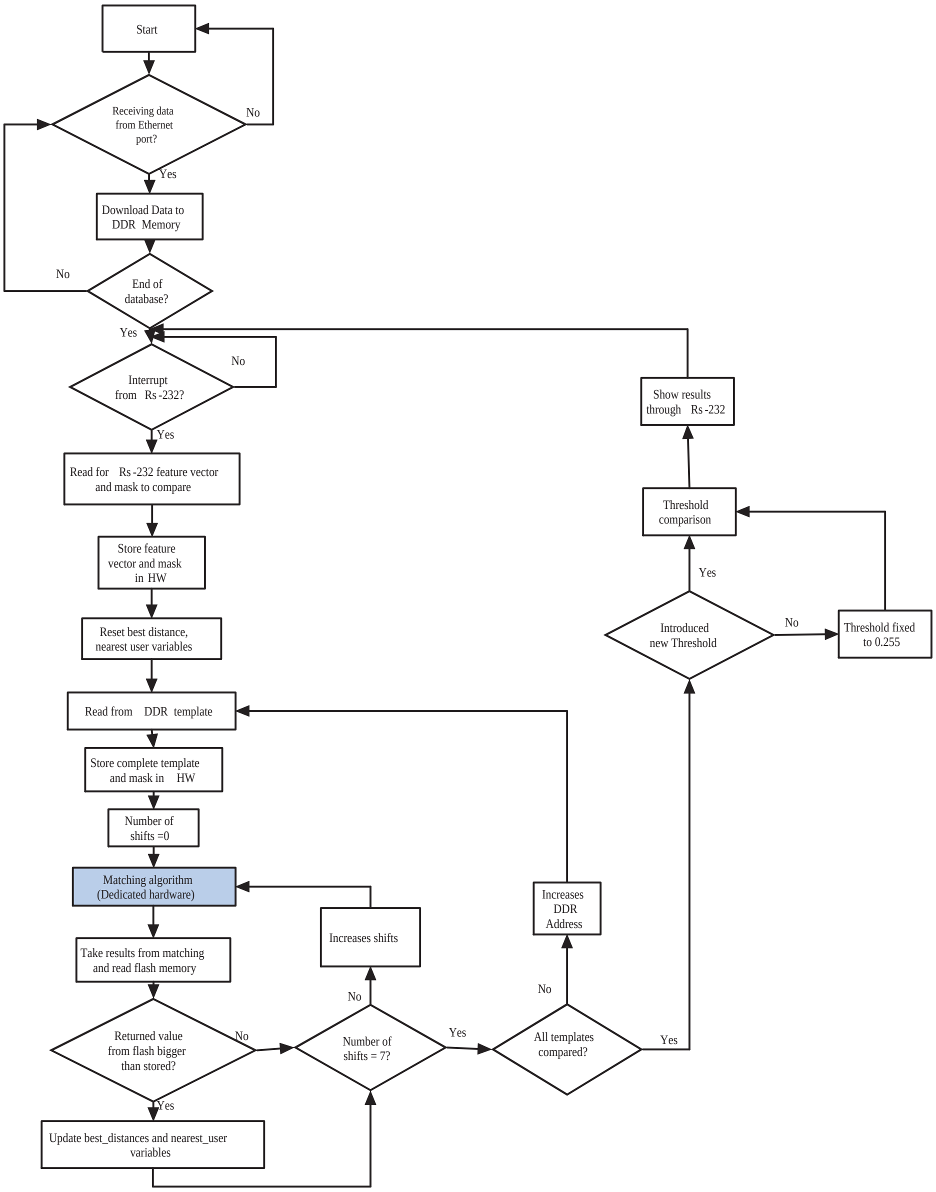

Start
Receiving data from Ethernet port?
No
Yes
Download Data to DDR Memory
End of database?
No
Yes
Interrupt from Rs-232?
No
Yes
Read for Rs-232 feature vector and mask to compare
Store feature vector and mask in HW
Reset best distance, nearest user variables
Read from DDR template
Store complete template and mask in HW
Number of shifts =0
Matching algorithm (Dedicated hardware)
Take results from matching and read flash memory
Returned value from flash bigger than stored?
No
Yes
Update best_distances and nearest_user variables
Increases shifts
No
Number of shifts = 7?
Yes
Increases DDR Address
No
All templates compared?
Yes
Introduced new Threshold
Yes
No
Threshold fixed to 0.255
Threshold comparison
Show results through Rs-232

Figure 11.4: Flow chart of the microprocessor during the search process

11.3.5 Implementation

As in previous experiments, we have used the Spartan3 XCeS2000 FPGA. We have employed the Microblaze as the central processor, which can be connected via the OPB bus to a wide range of different modules. Through this bus the microprocessor accesses the non-volatile memory, RS-232 interface, Ethernet interface and the SRAM memory. Hardware modules are connected using another faster bus, i.e. the FSL bus.

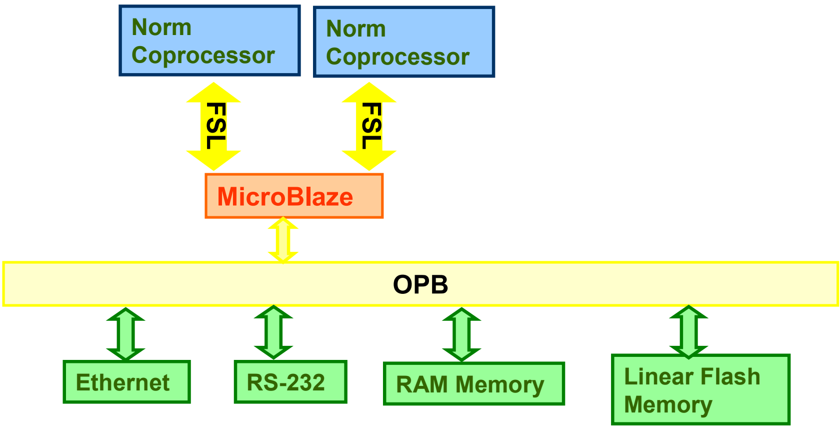


Figure 11.5: Implemented architecture

Figure 11.5 shows the microprocessor in Red , connected to 2 buses: OPB for most peripherals and FSL for dedicated hardware modules. Several peripherals are connected via the On-chip Peripheral Bus to the only one that the Microblaze 6.0 provides, i.e. the Ethernet and RAM memory. To connect the dedicated co-processors, a Fast Simplex Link is used. This bus is characterized by the speed it can achieve, as it provides a direct connection to the microprocessor, which is similar to a standard input/output without any additional protocol that ensures bus collision or error detection.

11.3.6 Analysis of the first architecture

Although the results obtained using this proposal were promising, further work has been performed on the system to attain improved performance. From this preliminary architecture two major bottlenecks have been identified, where both of these are related to the dataflow. The first bottleneck is arises due to the access to templates stored in the SRAM memory. When the dedicated co-processor is computing the comparison score, it should access the SRAM memory several times to read the templates and their corresponding masks. This action requires the co-processor to request this data from the microprocessor, followed by the microprocessor accessing the SRAM memory and retransmitting the requested data to the co-processor. Therefore, several clock cycles are lost during the data transfer due to the microprocessor participation. Each time the system must check for an identity, the dedicated hardware performs a number of accesses to the SRAM Memory given by 11.7:

$$accesses = 2 * n\_templates * n\_rotations * \frac{FeatureVectorLength}{RAMDataBusLength} \tag{11.7}$$

where *n_templates* is the number of templates stored in the memory, i.e. in the database, *FeatureVectorLength* depends on the algorithm used to obtain the vector, in our case, 256 Bytes; *RAMDataBusLength* is 32 bits in most situations, and *n_rotations* provides the number of rotations to be verified. Thus, this is the number of comparisons to be performed for a single template.

In this approach, each memory access takes 6 basic instructions and with the microprocessor working at 65.985MHz, 91ns are wasted in each access. Applying the abovementioned formula for 20 rotations and 100 templates, we conclude that close to 9 ms are unnecessarily lost. This time has been lost because of the microprocessor participation and does not consider the memory access time and hardware driver. Considering this, elimination of the microprocessor participation is found to be of major concern to minimize the access time.

The second bottleneck is also related to memory access but in this case, to the non-volatile memory access. This memory is used to store the look up table containing all the possible Hamming distance values. This memory was initially proposed to avoid the square root and division computation within formula 11.6. Thus, a complete template was examined and the numerator and denominator of formula 11.6 were computed, the data obtained was used to find the Hamming distance value. This solution was found to be faster than computing the square root and division, providing more accurate results as truncation and rounding have been avoided. As has been previously mentioned, in the case of massive identification, truncations can lead to significant errors which may produce incorrect identifications. By considering that the Hamming distance varies in the iris from 0 to 1, values close to 0 represent a positive identification and 0.5 for all other cases. Therefore, the results should be as accurate as possible.

11.4 Final architecture proposal

In this section, two solutions are proposed for the bottlenecks described above. These bottlenecks have lead to the development of a new architecture for the iris search engine. This new architecture has been optimized with respect to time when compared to previous approaches. This demonstrates its feasibility for use in border controls or stadium applications, also the reduced cost makes it suitable for human welfare applications.

11.4.1 New co-processor

The first bottleneck is solved by allowing the co-processor direct access to the SRAM memory, thus the microprocessor can be suppressed from this task. Several modifications studied in previous works have been made in order to implement this new idea. With these changes the co-processor will not only be composed of the matching co-processor, but also the SRAM driver and the additional logic used to perform control tasks.

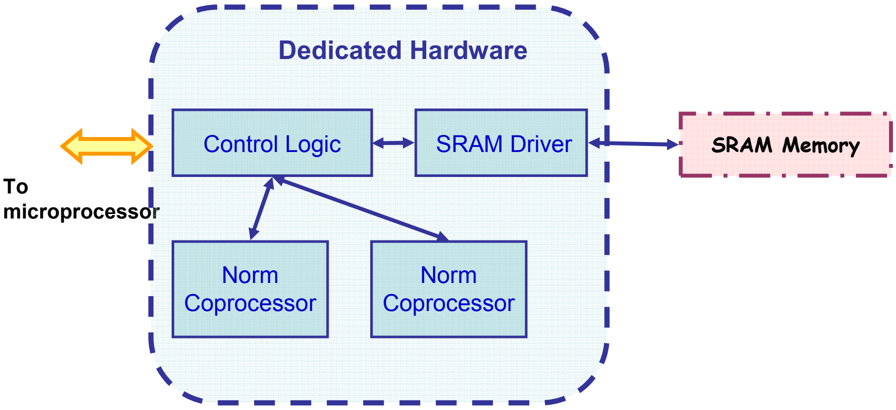


Figure 11.6: New co-processor architecture

This new proposal avoids unnecessary microprocessor participation in the matching process The matching co-processor, although based on previous approaches, should perform an additional function: the vector rotation. In previous versions, rotations were performed by the microprocessor in an intermediate state after reading the template from the SRAM memory and before supplying it to the matching co-processor. Since the workload of the microprocessor is reduced, dedicated hardware should not only perform the norm computation but also the rotations. Each template is rotated *n_rotation* times and, as previously mentioned, each rotation requires 16 bits to be moved each time. Since the memory data word length is 32 bits and depending on the rotation computed, the matching co-processor will ask for one or two memory words. To detect this, additional hardware has been implemented along with the matching co-processor.

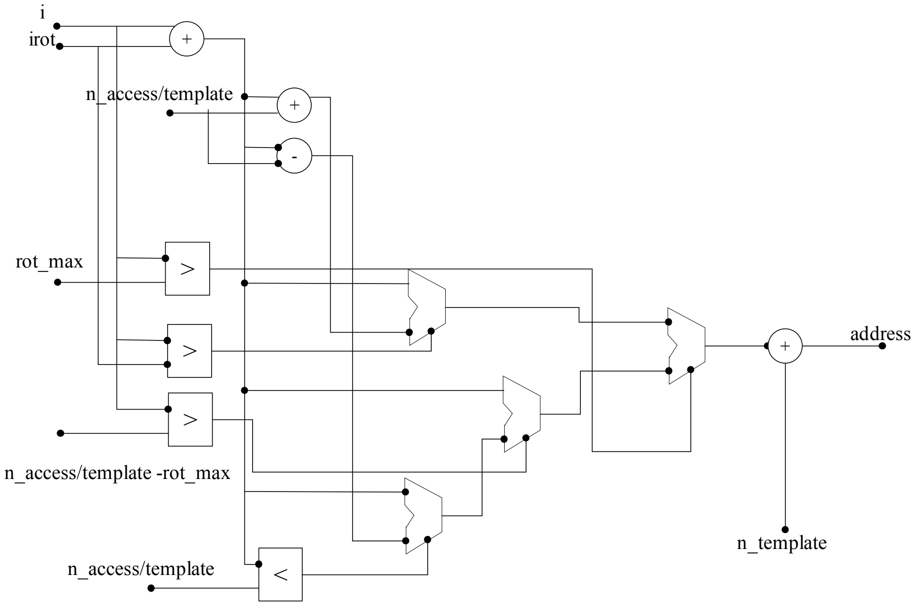


Figure 11.7: Additional hardware for address computation considering Rotations

Considering all the different rotations, the SRAM memory map is organized as shown in Table 11.1. This map attempts to reduce the computational time, this has been implemented by considering that sequential access is faster than random access. For this reason, authors have tried to sequentially store the data which is expected to be required at the same time as the matching process, thus, reducing the access time.

The control logic block introduced in the co-processor is used for several tasks:

- Controlling the Bus interface with the microprocessor: In previous approaches, the matching co-processor was connected to the microprocessor using a FSL (Fast Simple Link) connection [145]. This interface does not provide additional functions, such as interrupt controlling or bus arbiter. Additionally this is a non-standard link, thus portability to other platforms is not possible. For all these reasons, the new co-processor will be connected through an OPB (On-chip Peripheral Bus) interface [17], which provides the aforementioned functions and allows connection to other platforms.

Table 11.1: SRAM Memory Map

Memory Address	
0x00..0000	Fresh Sample Vector (First 32 bits)
	Fresh Sample Mask Vector (First 32 bits)
	Fresh Sample Vector (Following 32 bits)
	Fresh Sample Mask Vector (Following 32 bits)
	...
0x00..0080	Template 1
	Mask Template 1
	...
	Template 2
	Mask Template 2
	...
Last 20 Memory Addresses	10 Best results: Numerator, Denominator, user, Rotation

- A protocol has been created for this interfacing: The control logic block is in charge of transferring commands to the norm co-processor or to the memory and it should, when requested, report the state of the co-processor. This protocol is described in the following subsection.

- Controlling access to the SRAM memory: both microprocessor and matching co-processor will access this memory. This control logic manages the block which has priority in each computation step. In most cases the microprocessor will have priority except when the comparison process is being performed.

11.4.2 Protocol

As previously mentioned, connection between the co-processor and the microprocessor is carried out using an OPB interface. Such bus standard specify electrical and timing restrictions between signals which are part of the communications process [17]. However, there are no specifications on how to construct data or command frames. In our case study, establishing the difference between these two frames becomes interesting and, developing a complementary protocol using these structures is necessary.

The proposed protocol is as follows:

- Data frames: Data communication follows the plain OPB standard. Data of different lengths can be sent depending on the BE bits values. The memory address where it is required to be read or written should be specified, with data being transmitted via the data bus.
- Command frames: Accessing the first address of the peripheral is considered as a command transmission. When the master is performing a writing access, the data signal contains a command frame. On the other hand, when the master reads this first address, the slave writes as data, the state in which it is at in this moment.
  - *Master-Slave Commands:* The frame length is 32 bits, its structure is as Table 11.3 shows. The commands considered in our system are described in Table 11.4.

    Table 11.3: Structure of a Master-Slave Command Frame

Command	Data Length	Data	CRC
4 bits	5 bits	21 bits	2 bits

    The only command that allows data is the one related to the line parameter change. These parameters should be provided in the data field of the command frame.
  - *Slave-Master Frame:* when the master checks the state of the slave, it should read the first peripheral address. This peripheral will report its state in the data signal. The data frame returned is also formed by 32 bits.

    Table 11.8 summarizes the states considered in this proposal. Codes not considered are reserved for future use.

11.4.3 Restricted access to non-volatil memory

The problem when accessing this type of memory can be solved in a similar manner as in the previous bottleneck case. This solution, although faster, still requires continuous access to this memory each time a template comparison is computed. In order to reduce the number of accesses two considerations have been made: the first refers to the impossibility of the numerator being bigger than the denominator; and the second one is related to the Hamming distance behavior. Fig. 11.8 shows how the Hamming distance varies depending on the numerator and denominator values. In this figure, no restrictions have been considered.

Table 11.4: Commands from Master to Slave

OP Code	Command
0000	Start matching computation and obtain best results with data stored in the SRAM memory. Those results are stored in the last memory addresses.
0001	Stop comparison process.
0010	Continue comparison process.
0011	Allow database to be reloaded. This command stops the comparison process and allows a new database to be downloaded. If this command is not sent, any access to those addresses is not permitted except by the co-processor itself. In this way the user data is protected from undesired accesses.
0100	Database not accessible by the microprocessor, but still accessible by the co-processor. This command is mainly used when the database does not require all the memory space.
0101	Database not accessible either by the microprocessor or the co-processor.
0110	Change the line parameters for decision. The line, which helps to check which are the best results, can be changed according to application restrictions. For further details see following section.
Others	Reserved for future use.

Table 11.6: Structure of a Slave-Master Frames

State Code	State Additional Information	CRC
4 bits	26 bits	2 bits

Table 11.8: co-processor States Considered

State Code	Command
0000	Idle
0001	Downloading the database. Database downloading process is progress. Complete database is not downloaded yet.
0010	Database completely downloaded. Database still accessible.
0011	Database completely downloaded. Database not accessible.
0100	Downloading feature vector and mask for comparison.
0101	Ready for comparison.
0110	Computing matching. No results are available.
0111	Matching finished successfully. Results are available in the last memory addresses. Number of templates below the decision threshold is provided as additional information.
1000	Error in matching process. The ”additional information” field specifies which error has occurred: no data found, interruption in matching process, etc.
1001	Error in the last frame transmitted by the master. A required retransmission is requested.
1010	Command unknown, if the master sends a command the slave does not recognize as the same, it stops the process and goes into a waiting state until the master sends the correct command.

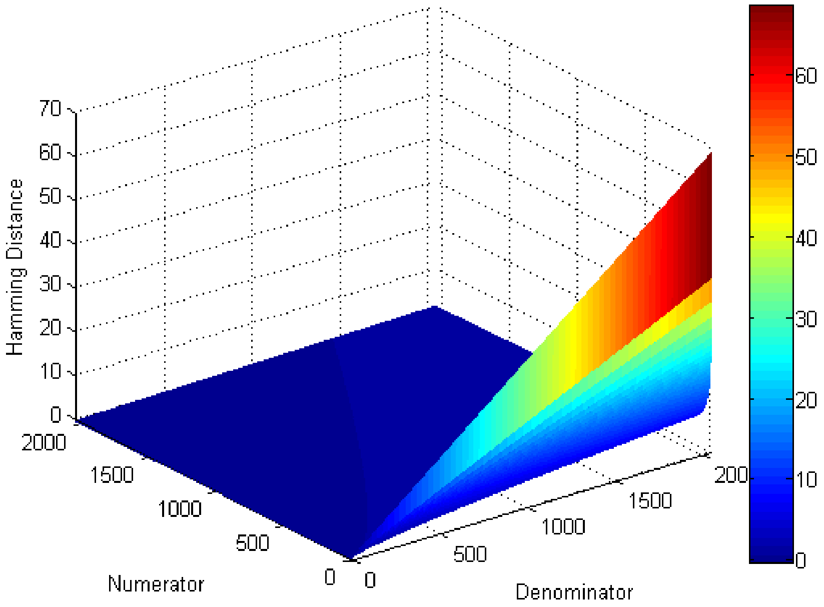


Figure 11.8: Hamming distance values depending on the numerator and denominator of formula 11.2

In formula (11.2) the denominator refers to how many bits of the feature vectors can be compared, as not all of them come from the iris region. On the other hand, the numerator checks which bits are equal by only considering those which come from the iris region examination. Therefore, the quotient of these two values should be below 1 (i.e.: numerator $<$ denominator): Daugman in [25] has demonstrated experimentally that the algorithm used for iris recognition achieves practically no errors when the comparison threshold is below 0.255. Fixing the threshold to this value, comparisons which lead to a Hamming distance above this threshold are considered to be matching with users different from that studied. As the target of a search engine is to find the identity of a user among those previously stored, the distances which are above this threshold should be discarded. Consequently, we can approximate this consideration as formula (11.8) shows:

$$(Th - 0.5)\sqrt{911} - (Th - 0.5)\sqrt{911} * Den \geq Num \tag{11.8}$$

where Th is the threshold, Den is the denominator 11.5, and Num is the numerator 11.4. We would like to remark that this approach is possible as the Hamming distance formula is continuous in the area of application, and does not show any breaking point in the region under study.

These formulae have been modified for their implementation so as to use a factor of 2 in all multiplication-based processes, and therefore, do not requireadditional hardware area or computing time.

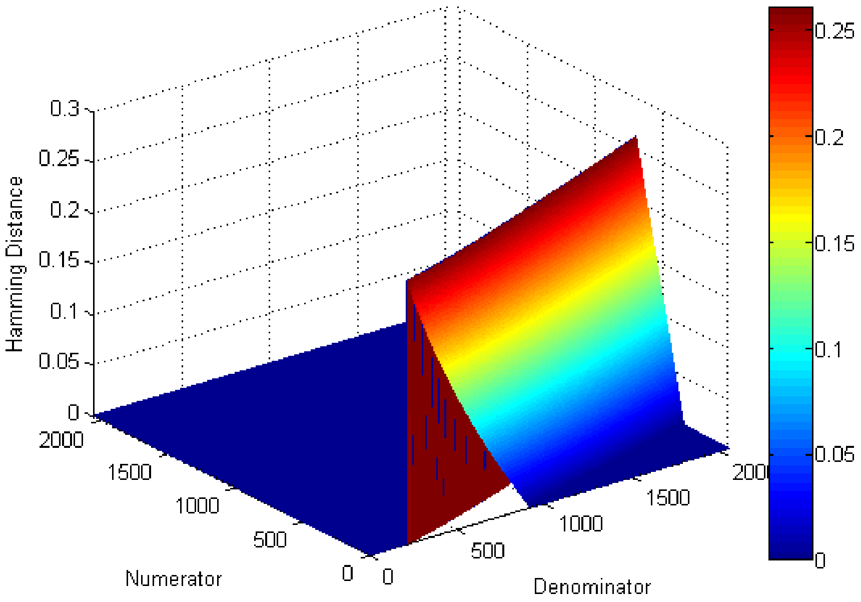


Figure 11.9: Hamming distance values suitable for comparison of a user with his/her template.

Fig. 11.9 shows how the Hamming distance varies with the numerator and denominator considering both of the previous restrictions. The graph has been rotated to appreciate the region of interest. In the blue color, the Hamming distances values are of no use because they do not satisfy the restrictions proposed, in different colors, variations of the hamming distance with suitable values of numerator and denominator, i.e. the region of interest. As it can be seen, if the numerator and denominator are in the zone delimited by those two straight lines, the resulting Hamming distance will be below the threshold, and, therefore, it is considered to belong to the user. By this simple method the number of accesses to the non-volatile memory is significantly reduced to those values with an expected Hamming distance which is below the threshold value. At the same time we have reduced the range of numbers which are to be stored, thus with the same word length represent a lower range, increasing the accuracy of the returned Hamming distances.

Additionally in this approach, the values stored in the non-volatile memory are previously quantified considering the probability of errors being committed [25]: intervals of quantified values vary from shorter, near the threshold, and increase as the Hamming distance get close to 0. By this non-uniform quantification, hamming distances closer to the threshold are more discernible than those closer to 0, therefore we avoid potential recognition errors as two completely different vectors lead to a higher hamming distance than two similar vectors.

11.4.4 Final search engine proposal

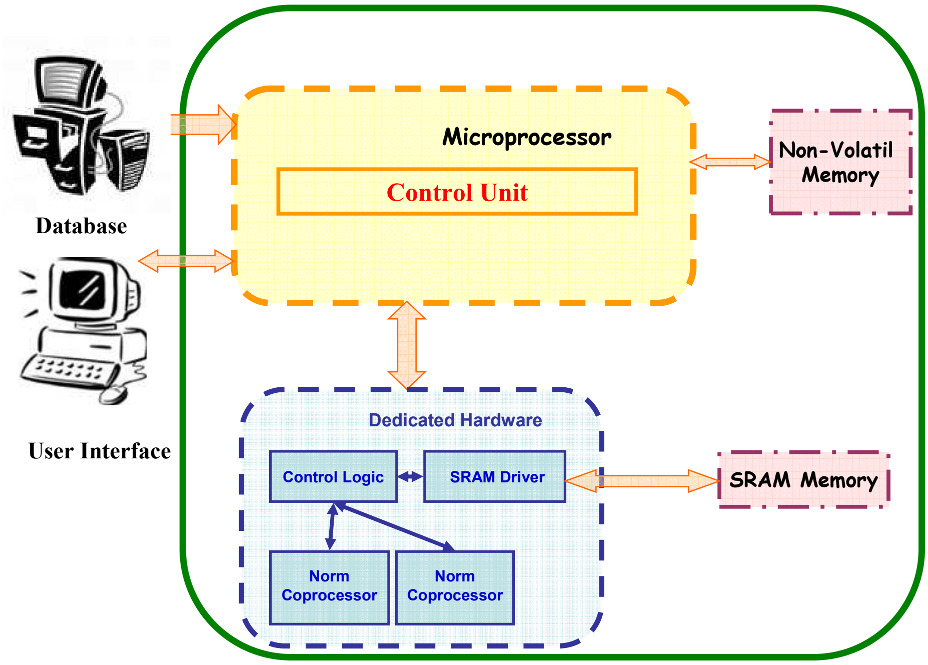


Figure 11.10: Architecture proposed. This architecture attempts to improve previous works, eliminating unnecessary communications.

These two modifications have lead to the new architecture shown in Fig. 11.10. As can be seen, several tasks have been moved to the new co-processor, these are rotations, matching and SRAM control, giving more importance to the hardware and attempting to avoid communication between the peripherals and microprocessors. The microprocessor in this new approach is in charge of the interface with the outside world which is required to download the database or to obtain the feature vector for comparison and to manage the system peripherals, except for the SRAM memory which is controlled by the IP processor created. The dataflow, in this system, is as follows:

1. When a reset occurs, the system goes into an idle state, until a download database command occurs.

2. The Database is downloaded to the platform through the Ethernet interface. Once this database is stored, the system will wait until a sample and its masks are provided for comparison.

3. The Sample vector and its mask are stored in the SRAM memory as a vector length and does not allow storage of the same vectors in two shift registers of the hardware.

4. Initialize the number of templates to be examined.

5. Initialize the number of rotations which are to be performed.

6. The norm co-processor will start the matching process, this requires access to the SRAM memory via the control logic implemented in the hardware. The reading of addresses depends on the rotation considered for each case and the template being examined.

7. This co-processor will return the numerator and the denominator of the template rotated. The data is verified to belong to the area delimited by the straight lines shown in formula (11.8). If the data obtained is inside this area it is stored in the last addresses of the SRAM memory.

8. If all of the rotations are not checked, the rotation number is increased and goes to point 6).

9. If we have not checked all the database users, we go to point 5) changing the template which is to be examined.

10. If all templates and rotations have been checked, the co-processor sends a finishing signal to the microprocessor, which will now access the last memory addresses, where the best results have been stored.

11. This data is verified in the Look-Up Table memory to find the smallest distance.

12. The system will send the user operator a message informing of the best results obtained. If no results are stored in the SRAM memory, the system will send the user a message indicating that the user does not belong to the database.

Steps 4 to 10 are performed using dedicated hardware instead of using the microprocessor which was used in the first approach. As can be observed, the computational cost is translated largely to the dedicated hardware, releasing the microprocessor's functions; increasing in this way, the speed of the complete system and avoiding unnecessary data transfers.

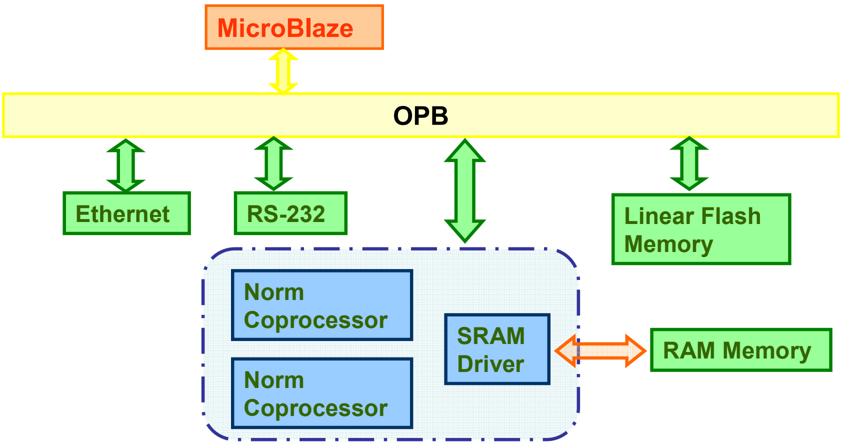


Figure 11.11: Architecture proposed.

Several differences can be observed from Fig. 11.5 and Fig. 11.11. As previously mentioned, the co-processor now is formed by two norm co-processors, the same element used in the first approach. However, the connection of these co-processors is done via an OPB, thus avoiding the use of the non-standard FSL. The new peripheral is considerably complex as it is formed by the two previously mentioned co-processors, the SRAM driver and additional control logic to interact with the microprocessor following the OPB protocol and controls access to the RAM memory from the co-processors and the microprocessor. The most relevant peripheral is the RAM memory whose driver is not now implemented in the microprocessor, but instead as dedicated hardware, this allows direct access to it by the two dedicated co-processors.

11.5 Results

Several experiments have been carried out to demonstrate the feasibility of our proposal. In all these tests we have compared results with those obtained from previous approaches, this demonstrates the improvements achieved in this work. All tests have been done within the same platform and database to obtain an effective comparison of both architectures. All these experiments study different parameters which influence the complete system cost: hardware area and memory size, and demonstrate the vaibility of our proposal for commercial implementation

11.5.1 Test conditions: Database

As regards the database used, it is formed by 100,000 iriscodes from different users. These iriscodes are obtained from real potential users, not artificially [158] or randomly. The iriscodes come from different people of different races, and are part of the database employed in [25]. The database has been divided in several small databases to check the influence of the database size on several of the aspects which have been studied.

11.5.2 Area

As regards the area of the hardware, we have studied two different aspects: the FPGA hardware area occupied and the memory size required for both implementations. The area of the occupied Hardware influences the FPGA where the system is to be implemented. Low cost FPGAs are desired for cost and size reasons. The memory size not only influences the area of the system, but also its cost. To date, memory size per unit area is quite small and is constantly decreasing, however, its cost increases with storage capacity.

11.5.2.1 Memory Size

Two memories have been used in the system: an SRAM memory and a non-volatile memory. These two memories are analyzed in this section to check each required size in both implementations.

- SRAM memory: This memory is used to store the database in previous approaches and to store the database and best results in the approach proposed in this paper. The tests presented in this section have been carried out using a memory word length of 32 bits . Table 11.10 presents the SRAM size which depends on the number of users contained within database, considering that each user requires a 256 byte iriscode and 256 byte mask associated with this code. If the number of users is low, the difference between the two approaches is noticeable, as the number of bytes dedicated to store the best results is similar in range to the database bytes.

  Although Table 11.10 is clear, the actual situation is not as this table shows: commercial SRAM gives their storage capacity in KB (Kbytes), i.e. multiple of 1024 bytes. Considering different sizes of SRAM memories the number of users does it can store does not vary from one architecture to the other as SRAM sizes

Table 11.10: SRAM Size (Bytes) depending of the number of users the database contains

Number of users	20	100	5000
Initial Approach	10240	51201	2560001
Final Approach	10321	51282	2560081

do not allow storing an exact number of users, leaving in the first approach free memory which is used in the second one to store the best results.

In case the database cannot be stored in a commercial SRAM memory due to its size, a parallel architecture is recommended. Dividing the database in several parts and storing each of them in a different search engine. When an identification process is requested, all the search engines work simultaneously with each of the database parts, providing results in less time than the case where only one one search engine is used [86].

- Non-Volatil Memory: This second memory has been used to store the Look-up table in both approaches, thus it is independent of the database size. This size has also been modified from the previous approach and it is fixed by the number of possible values that the Hamming distance can have. In the first case it is composed of all the possible values and in the new approach it is reduced to those values which are closer to best possible values.

Table 11.11: Non-Volatil Memory size

	LUT Size	Memory Used
Initial Approach	4 MB	4 MB
Final Approach	968 KB	1 MB

As in the previous memory, the size of the memory is set by market solutions, Table 11.11 shows the Look-Up Table size and the Memory size used to store this look-up table. It can be seen that a reduction of 400% has been achieved, reducing the cost associated with it. At this point it is also important to note the advantage introduced from the second approach on the accuracy achieved. In both cases, we are dealing with a memory word length of 32 bits; in the former approach, the range of numbers to store is from 0 to 70, while in the current proposal the variation interval of the Hamming distance is from 0 to 0.25. With the same number of bits, in the second case, the number of possible values to implement is higher, increasing the system accuracy, which in return translates to lower amount of errors.

11.5.2.2 FPGA Area

As previously mentioned, an FPGA is formed by several elements; in the case of the Spartan 3, these elements are slices, block RAMs and Multipliers. Slices are formed by lookup tables and flip-flops, used for implementing sequential and combinational processes; Block RAMs are used for internal memories of the system and for input-output purposes; finally, multipliers were introduced due to the high demand for this operation in many applications.
In hardware occupancy the database size has not been considered as the FPGA area is independent of this parameter.

Table 11.12: Co-processor hardware area

	Slices	BRAM	Mux
Initial Approach	90(0%)	0(0%)	0(0%)
Final Approach	329(1%)	0(0%)	0(0%)

The dedicated hardware used for the co-processor implementation is pointed out in Table 11.12. The current approach shows a significant increment in the area used, as it has included the SRAM driver and additional control logic. Moreover, two old co-processors are included in the new proposal as both the numerator and denominator are computed using the same device, this was not carried out in previous approaches. For the implementation of the co-processor no Block RAM (BRAM) or Multipliers have been used due to their functionality. Table 11.13 presents the complete area required by both approaches. The difference which exist, shown in the previous table, is not as significant as before. Although the best results in area have been obtained from the first approach, the current approach has not increased significantly the slices used, increasing it by just 1%. However the values are quite low and both solutions can be implemented on similar devices.

Table 11.13: Complete System hardware area

	Slices	2 BRAM	Mux
Initial Approach	1266(6%)	32(80%)	3(7%)
Final Approach	1522(7%)	32(80%)	3(7%)

11.5.3 Time

Although the area results have a great influence on the performance, cost and time computations are probably more important. High computational time is not affordable for these systems due to the impracticality of having a user wait. In this section, time results will be analyzed to check if the proposal developed is capable of performing identification faster or in a similar time than other approaches.

Table 11.14: Time Comparison based on the users for 20 rotations per user

Number of users	20	100	5000
Initial Approach fmax =66.716 MHz	52.43 ms	262.18 ms	13,109.15 ms
Final Approach fmax = 62.637 MHz	5.46 ms	27.32 ms	1,365.01 ms

As Table 11.14 shows, the time has been reduced significantly, although the frequency between both solutions has not. The microprocessor participation in the reading and/or writing process in the SRAM memory increases the computational time, as mentioned in Section III. When compared to software approaches [26], the computation of 1000000 comparisons takes around 1s in a machine composed of 6 parallel microprocessors at 3GHz with a mean of 3 rotations per user. Our implementation has not been tested for the aforementioned amount of data, but by extrapolating our results, we achieve close to 41s with the current solution and 410s with previous one at the frequency set in our tests. Thus, if we consider 6 platforms in parallel, to compute one million comparisons under the same test frequency conditions, our solution will require less than 7s. Furthermore, considering just one single platform at 3GHz, it will take 0.8s, and 6 platforms in parallel at 3GHz, 0.14 s, a much more efficient result than the software solution. Although these results are unachievable due to the SRAM time access restrictions, it would be possible if we replaced it by a DDR RAM memory module, where the access is much faster. However, these memories require drivers to perform further functions, such as refreshing, thus increasing the clock cycles required.

11.6 Further proposals

Both architectures mentioned in this chapter present a similar problem: both of them are unable to store large databases due to memory restrictions. When the database to be checked is larger than the SRAM storage capability, other solutions are required. In this section, we will introduce a proposal for this problem based on the architectures described above. Large databases entail high levels of time consumption to check all the stored users. To deal with this problem, we recommend a distributed architecture where several matching processes can be computed simultaneously, this thanks to concurrence. Essentially, the dataflow used for this approach is described as follows:

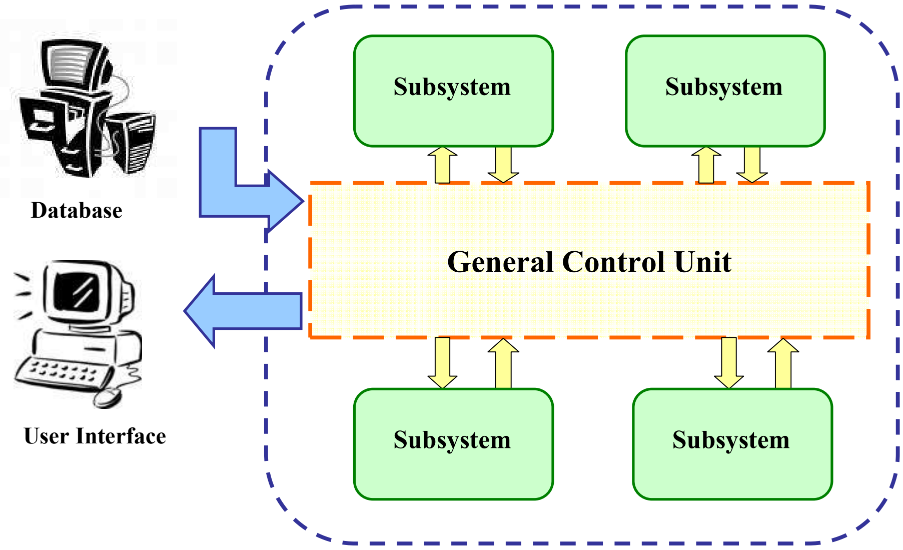


Figure 11.12: Larger database approach.

the database is split into several parts for the downloading process, each of them into a separate subsystem. When a comparison for a new sample vector and its masks is desired, it is transferred to each of these subsystems, and these perform the comparison process by applying different parts of the database simultaneously. Once all subsystems have finished, the best results obtained from each of them are returned to the general control unit which chooses the best result obtained from each subsystem as the potential holder of the sample examined. All subsystems are controlled by a central control unit, this is formed using an additional microprocessor which performs the following tasks:

- Identify which of the different subsystems is working on the database downloading process, so when database transfer is occurring, it should control which of the RAM memories is being addressed. During the comparison process, all subsystems can work simultaneously.

- This microprocessor should be the only element with the RS-232 interface, this avoids multiple accesses to this resource. Once the feature vector and mask are provided, it should be transmitted to all the subsystems. When the computation is finished, all subsystems should return their best values to the general control unit. This unit will decide which, out of all considered, provides the best result, returning it to the user though the serial interface.

- Only one single Ethernet connection is required. The central microprocessor will deal with bridging data transfers to the corresponding subsystem.

As regards the subsystems, depending on the desired approach to be used, these may have different architectures. If the initial architecture is chosen, the subsystem should be formed by a platform similar to this architecture, i.e. using a microprocessor, two norm co-processors and RAM memory. On the other hand, the microprocessor is not required if the second approach is used. The microprocessor is necessary in the first case as a peripheral arbiter between the RAM memory, co-processors and the central control unit. However, in the second case, the co-processor control logic performs this function, reducing the whole area and thus, system cost.

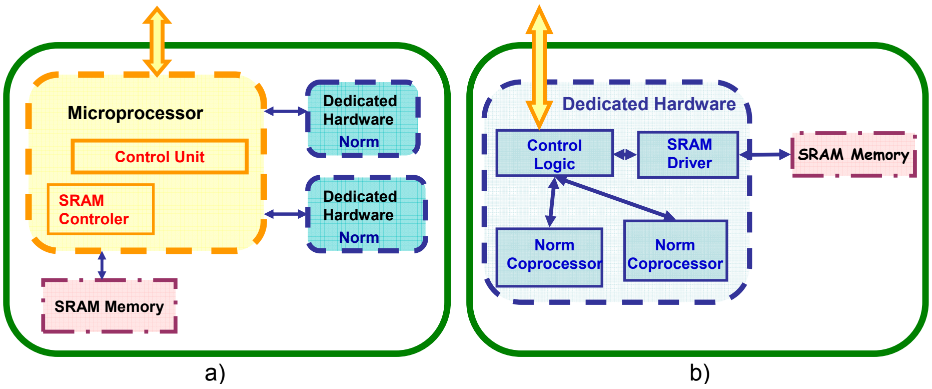


Figure 11.13: Subsystems for large database approach: a) Using initial architecture. b) Employing final approach.

11.7 Conclusions

We have presented in this chapter two possible architectures for an iris biometric search engine. Both proposals are based on hardware/software co-designs, but the main difference is in the connection of elements. The first proposal has relied on a central-distribution, where a main microprocessor works as a central connection unit that controls each of the accesses. A dedicated co-processor has been designed to perform the biometric tasks, i.e., in this case, comparisons. These comparisons require continuous access to the memory, and therefore, central microprocessor participation. Considering that this procedure increases the processing time, the second proposal has been designed to redistribute the elements accesses between the microprocessor and the dedicated co-processor, providing direct access to the memory and thus reducing the computational time. Another important difference between the two proposals is the non-volatile memory used. In the first approach, this memory is addressed n_rotations times for each template, whereas in the second approach the addresses are reduced by a factor of 10. This has been possible because of intelligent value discrimination provided by an in-depth study of the Hamming distance distribution. Several tests have been performed using both architectures to check the main differences, advantages and disadvantages. All these tests have clearly indicated that by using adequate system designs the performance is improved in several ways, such as time or memory consumption. Another advantage to using both proposals presented when compared to conventional solutions, i.e., software based solutions, is the security improvement provided by a hardware/software co-design. Thanks to these implementations' cost , search engines can be located at each access point as required; avoiding iriscode transfers via an internal or external network, thus reducing possible attacks. Both approaches cover several situations where massive identification is required such as in border controls which have no means of identification and access to large facilities and buildings. As tests indicate, the temporal requirements for these situations has clearly been covered, while at the same time, a reduction in the area leads to more cost effective solutions when compared to software implementations. Therefore, the architectures presented here have demonstrated, by means of several tests, their implementation feasibility and use in several situations where cost and the time to carry out the identification process should be as low as possible, and also where the security levels of the whole system are increased.

Part IV

Conclusions and Future Work

Chapter 12

Conclusions and future work

This Thesis has provided an in depth study and experimental results on the contributions hardware/software co-design offer biometric identification systems. A successful attempt has been made in this dissertation to cover all the different fields of investigation by employing many different techniques to verify their viability. An overview of Biometrics has been presented, providing a specific focus on the latest developments in Iris Biometrics, also, in the second part of this Thesis, the Hardware/software environment has been described. The state-of-the-art of Iris Biometrics and hardware/software co-design can be found in chapter 3 and chapter 6, respectively. Other important issues, such as the security of Biometrics and ID tokens, have been addressed where this is complimented with a study of the contributions made in this area.

In this Thesis, two important contributions have been made to the Biometric identification process: the first, verification of a system based on ID tokens and secondly, a detailed description of a search engine capable of performing massive recognition, both of these contributions have been considered for Iris Biometrics.

The first relevant contribution is a Biometric System architecture proposal based on ID tokens in a distributed system. In this contribution, we have specified several system considerations and have presented the different tasks of each element forming it, such as central servers and/or terminals. The main function of the terminal has been carefully designed so that its principle and only task is to acquire the initial biometric raw data, which is then transmitted, using security cryptographic methods, to the token where all the biometric process are performed. The ID token architecture has been based on a hardware/software co-design. The architecture proposed, which is independent of the modality, divides the biometric process into hardware and software so as to achieve further and increased function performance when compared to existing tokens. This partition has not only considered the reduction in computational time that hardware structures provide, but also the reduced area and thus power consumption, the increase in security levels and effects on the performance of the complete design.

To verify this proposal, we have implemented an ID token based on Iris Biometrics following the aforementioned issues. We have developed different modules for an iris algorithm both on hardware and software platforms and have obtained the results required for an effective combination of both methods.

We have also studied different alternatives to solve the partition problem which exists for hardware/software co-designs, leading to results that have indicated that a tabu search is the fastest algorithm for this purpose.

Finally, using all the data obtained, we have been able to construct different architectures which are dependent on the required constraints. These architectures have been presented considering time as the principal constraint and we have obtained a reduction of as high as 30% in the processing time when compared to all other software solutions. This solution has also provided the least amount of area and power consumption. By considering the performance as the chief constraint, two different architectures have been presented, the first also attempts to minimize the processing time and the second reduces the hardware area and power consumption. With respect to the security we have also demonstrated two different architectures which consider both the time and hardware area as secondary requirements. Finally, the last architecture presented considers all these different design constraints to be of equal importance.

These architectures have allowed us to study the impact hardware solutions have on the security of the system against possible authentication attacks, how the performance is influenced by the lack of floating point operations in hardware modules and how hardware solutions reduce the processing time with respect to software solutions, which reduce the hardware area and thus, power consumption, etc.

The second contribution that made has been the development of a search engine for massive identification applications, here time is a major constraint as the comparison should be performed over millions of users. We have initially proposed an implementation based on a hardware/software co-design where the memory is connected to the microprocessor, although the comparison has been performed using a dedicated hardware co-processor. In a second approach, the memory driver has been connected directly within the same hardware coprocessor as in the previous implementation; therefore, the dedicated hardware can access the memory without the intervention of the microprocessor. This architecture has demonstrated the importance of the connections between the elements used when time is considered as the major constraint. Both architectures have been compared in time and area to provide the final conclusions.

12.1 Future work

12.1.1 Motivation for future work

The research performed in this Thesis has allowed us to identify the existing requirements for innovative biometric applications that guarantee maximum security and efficiency for everyday activities. Keeping this in mind, here we comment briefly on the relevant aspects related to potential requirements which are directly related to our conclusions. We are convinced that the proposals made in this Thesis will provide new techniques to face the challenges of Biometrics in the 21st. century.

The new EU citizenry & border security measures

Ever since the 9 11 attempts suffered by the United States in 2001, security measures regarding passage of citizens from one country to another has been of major priority. Almost a decade after and in lieu of similar attacks suffered in Madrid and London, the European Union has progressively enforced stricter measures both at entry control points to the 27-Member State territory as well as to the permanence of illegal foreign nationals already within EU countries.

The structure of the Schengen Agreement, in vigour in 2009, has elevated to a maximum priority the consideration of sophisticated biometric applications to restrain security breaches. However, four EU Member States are partially excluded from the application of Schengen regulations (United Kingdom and Ireland voluntarily, while Bulgaria and Romania have been suspended until 2010), three EFTA countries (Norway, Iceland and Switzerland) offer free immigration flow from other continents into the other 23 EU countries.

Due to the surge of illegal immigration, particularly at the principal airports of the 25 full-pledged Schengen countries, the European Commission has proposed stricter measures at border controls. Such measures are being accompanied by several experimental actions, one of which will be the application of biometric passports in Spain's international airports (Barajas, Madrid and El Prat, Barcelona). In Spite of the United Kingdom and Ireland's opting out of several of the border control arrangements included in the Schengen Agreement from the Amsterdam Treaty, these countries actively participate in judicial and police cooperation under the scope of Europol and the G-5 security group.

Public & private enterprise biometric security requirements

Both public and private enterprises each day demand improved and increased security measures to ensure the safety of their industrial and commercial activities. Biometrics has become the key source of such progressive improvement of global security at premises and installations with ever-increasing protection requirements. Furthermore, both government buildings and private offices have demanded optimum security not only for physical sites but also for intangibles such as information, know-how, patents, designs, etc. which are stored in such installations.

With the globalization of business networks, the need for biometric security measures in public and private installations increases. Curiously enough, the most popular biometric trait in use, the fingerprint, has been put to the test using sophisticated attacks both on public and private enterprise systems. In Spain, the use of digital ID is being encouraged to minimize forgery attempts of official government documentation as well as its use in regular banking operations. The aforementioned ID is currently being used as a model for other similar applications within the rest of the EU Member States, particularly for the implementation of the first ID documentation within the United Kingdom. Iris Biometrics will most probably have the final say in near future developments of such identification techniques.

The enormous and increasing presence of unregistered foreign nationals within the EU requires constant control solutions that can effectively resolve security issues faced both by Public Administrations in different Member States and major business enterprises with numerous employees located in different parts of the territory.

Iris Biometrics in Global 2.0 IT security measures

In a globalized world where Internet has torn down all types of communication barriers between nations, continents and individuals, it is impossible to imagine returning to the traditional methods of information transfer.

What has often worried developers of IT tools has been the incorporation of security measures to guarantee secure information transfer, privacy protection of personal information and the elimination of identity usurpation in transactions or identification requisites.

While all conventional measures have long been discarded and even the most popular biometric applications (fingerprint) are been questioned, the use of Iris Biometrics is at the threshold of its numerous possibilities for everyday 2.0 IT uses. This challenge is standing before us at a time when this Thesis has demonstrated a means of living up to the expectations in a short-medium term.

12.1.2 Proposed future work

One of the main targets of research is to seek answers to questions, finding better solutions to something previously accomplished. During these years of research, performed for this Thesis, several questions have been made where the majority of them have been solved however, several remain unanswered. All these questions have led to well defined future work on themes related to the conclusions of this Thesis; further efforts which will lead to new improvements in biometric systems. The different ideas which have arisen during these years, but have not been carried out, have been detailed in this final chapter. Due to the interdisciplinary nature of this Thesis, many different future areas of research can be faced all of which contain an extensive quantity of topics:

12.1.2.1 Image processing

As part of our proposal for an optimized ID token design, we have made use of an algorithm proposed by Sanchez-Avila. This algorithm has been studied in depth and several modifications have been proposed to improve its performance and reduce the error rates.

Image quality determination

The first proposal made refers to the quality of the input image. The algorithm implemented demonstrates improved results when high resolution images are used thus, determining the quality of the input image at preliminary stages of the identification process will aid in rejecting or accepting images depending on the probability of an error occurring.

Several facts should be considered when determining the image quality, such as image size or iris size. However, one of the most problematic factors is due to blurring. Several different techniques have been devised to detect if the image is focused, these are [74]:

- Daugman's proposal: Daugman in [22] has proposed a method to determine the image quality by studying the total power in the 2-D Fourier domain for high spatial frequencies. Instead of using a 2-D FFT, which is highly computational, use can be made of a convolution kernel along with an emphasis filter where only integer arithmetic is used.

- Entropy computation: Entropy measures the amount of information contained within a signal. If we compute the entropy of the initial image we obtain a relatively high entropy value for high resolution images; otherwise, it would stand for pixels of a similar value, i.e., there is limited information provided by the image, thus the image is unfocused.

- Edge detection: blurred images do not present clear edges, thus by examining the edges the blurring can be detected.

Pre-processing

One of the main problems we have detected in this algorithm is located in the pre-processing block. This block detects the pupil using morphological operators, however this is not performed on the outer iris boundary, as it is not necessary for the following steps of the identification process. However, in many cases, obtaining this boundary is necessary to avoid errors being committed in subsequent identification stages. Consider, for example, the image shown in Fig. 12.1.

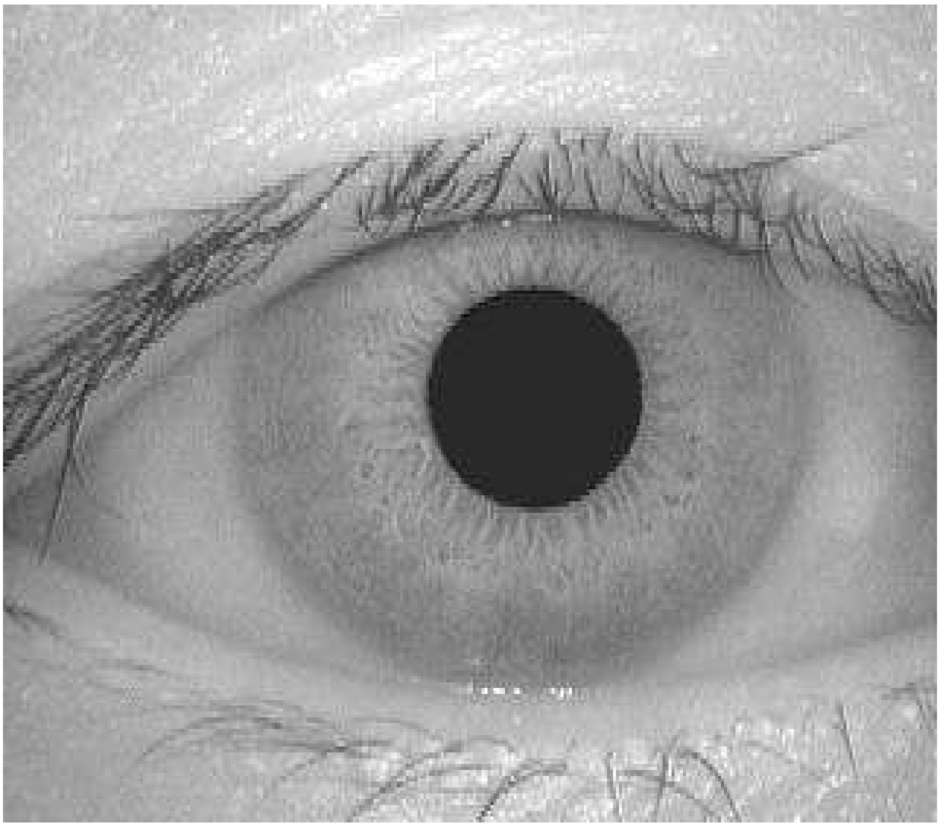


Figure 12.1: Image from Casia data base

It may be observed in this image that the iris is partially occluded by the eyelids and eyelashes thus, when the annulus is taken, part of the eyelids are also taken, and though the bits which come from that region are of no use, they may lead to an error.

This image is not a unique case; in fact, this situation (part of the iris occluded by eyelids) typical occurs in eyes from Asian people, where eyelids may be observed to overlap the eyes significantly. A similar situation occurs with eyelashes, these provoke errors as they can be confused with the ridges of the iris.

We have attempted to avoid this situation by detecting the outer boundary using a similar morphological process that is used on the pupil; however, the results obtained were not satisfactory as the outer boundary is not as well defined as the pupil-iris boundary. As a result, a new pre-processing algorithm is required.

As well as the aforementioned problem, the former pre-processing algorithm presents a further limitation with respect to its approximations. In the pre-processing algorithm it is assumed that the pupil and the iris are circular, however, in some cases, they are not. The camera angle or certain illnesses may lead to non-circular pupil images. In such cases, the pre-processing block is also observed to fail.

To gain system universality, other relatively flexible pre-processing alternatives must be considered to detect both boundaries. Amongst all the different alternatives mentioned, active contours stand out. This tool is widely employed in image vision to detect shapes within images. This tool is based on the location of curves within images which vary in shape according to forces, until they obtain a state of minimum energy. What makes this technique interesting is that these particular forces depend on several terms. Several rely on the capabilities of the initial curve or, what is even more interesting on the image where these are situated. Using this solution, an initial circle can be situated within the pupil and then, considering the edge map of the image, the form of the pupil can be detected. By considering a power map which increases from the pupil to the following outside edge, the outer boundary of the iris can be established.

Using this tool, we not only control non-circular pupils, but also eyelashes and eyelids can be detected, as the deformation of the curve is determined by the designer's parameters. Thus, the curves can be made as elastic as required, even to the point of detecting discontinuities on a line or at sharp corners.

The main problem associated with active contours is the high level of computation required to find the minimum energy, as the energy computation is generally performed using exhaustive search. However, using heuristic solutions, such as those presented in the research of the minimum partition function, the computation required is significantly reduced.

Dynamic feature extraction

Out of focus images have a significant influence on the feature vector, and thus, on the final results. Feature vectors have been obtained as the zero-crossing representation of

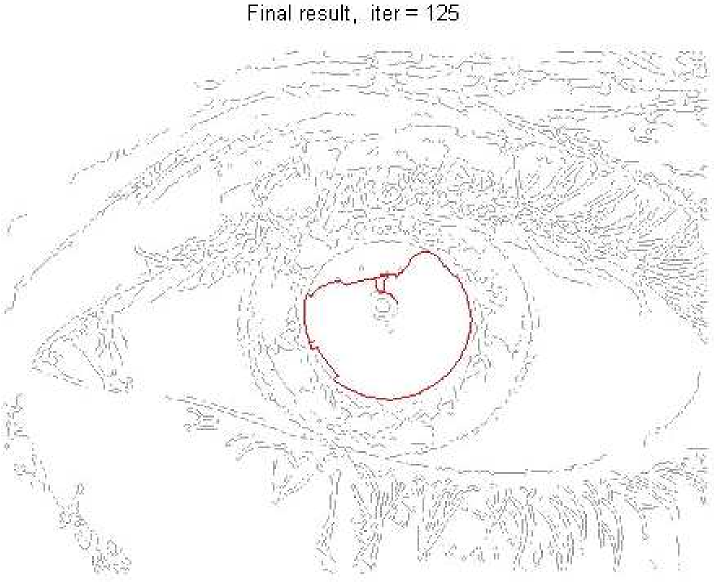


Figure 12.2: Iris Segmentation using active contours

a wavelet transformation of an annulus around the pupil. The Wavelet transformation is highly sensitive to the initial image resolution, especially regarding the scale considered to acquire information: The higher the resolution, the higher the scales to be considered are.

In our algorithm, we have computed a wavelet transformation using 8 scales, but for the feature vector we have only considered scales of 4, 5 and 6 which are independent of the initial resolution, as experimental results have indicated that these scales contain relevant information required for pattern recognition on high resolution images. But what happens when the initial images do not have a high level of resolution? In this case, the information is located in lower scales. Our proposal is a straightforward technique to extract the dynamic feature vector, based on this fact and the aforementioned resolution computation, and is performed by simply considering different scales according to the resolution of the initial image.

This idea of dynamic acquisition of data can be extended to other algorithms, such as Gabor Filters, where the parameters may vary with the resolution to obtain improved information.

Aliveness detection

In this document we have discussed a solution to prevent authentication attacks. However, no countermeasures have been considered to concealing attacks, i.e., using artefacts or artificial samples to access the system. These attacks are typical for biometric systems, and depend on the trait studied. For iris recognition, we can divide the potential concealing attacks into the three following categories [24]:

- 1. Photonic and spectrographic countermeasures: These rely on the specific manner in which tissue, fat, and blood interact with light of different wavelengths, in particular at near-infrared wavelengths; a similar situation occurs with melanin pigmentation, both for the skin and the front layer of the iris. Among the different photonic countermeasures, we may also consider the use of coaxial retinal back-reflection (commonly called "red-eye" effect), and the 4 Purkinje reflections which occur as a result of the eye's four optical surfaces (front and back of lens, and front and back of cornea).

- Behavioural countermeasures: These can be divided into two different categories, involuntary and voluntary, the former is related to the autonomic nervous system and hence not consciously controllable by the user, i.e. changes in pupil size, whether or not the light level is changing, the latter describes conscious control, such as challenge responses, i.e. eye movements or blinks on command. These tests ensure that the brain is connected to the eye.

- Countermeasures against analogue physical attacks, such as fake iris patterns printed onto contact lenses: These can be detected by searching for the dot matrix characteristic of several printing processes, these can also be detected using photonic techniques on the printing dyes, also the fact that a contact lens sits on the cornea producing a curved spherical surface located at the front part of the iris, whereas the iris itself is a relatively flat internal organ within the eye. Also, the position of bright reflections from the cornea are expected to change when the light comes from different angles, however there is no change in angle when a photograph is illuminated.

Several algorithms for aliveness detection have been proposed and involve the modification of the lighting during the identification process. Another technique involves a recording of a short video file instead of acquiring a single photo, the movement of the pupil is compared with a theoretical model of the human pupils movement under the same lighting conditions. Active contours can also be used to detect movement thus, aliveness detection would be possible with the modified pre-processing algorithm [112]. In a similar way, the detection of a user under duress can be achieved by modelling the pupil's movement, thus detecting any potential social engineering attacks.

12.1.2.2 Further biometric tokens

Several areas of possible future work, closely related to this thesis, rely on the improvement of ID tokens used for identification, i.e. considering other platforms to implement biometric devices. Considering this concept, we highlight possible areas of future work which are based on the construction of improved and more secure biometric systems.

12.1.2.3 Terminal development

This thesis has begun with the development of a methodology that considers several facts surrounding the ID token. However, it must be kept in mind that these tokens are connected to a terminal. The development of these terminals must be carried out using the same principles as the tokens described in this thesis, i.e. not only considering the performance and/or time, but also the cost and security. For such considerations and for the particular case of iris recognition, the terminal must be formed by a camera and additional control logic, following is a discussion on each of these elements:

Iris camera

Currently, a lot of research effort, within the group where this Thesis has been developed, is being focused on the use of new Iris cameras. A model based on fixed field depth has already been implemented, and further research is being carried out on the detection of the correct field depth according to the acquired images. The resolution should be computed in the additional control logic thus, if the image acquired does not satisfy the resolution requirements, another image can be acquired.

Additional control logic

Besides the control of the camera and the computation of the resolution, the control logic in the terminal should perform other tasks related to the transfer of data between the terminal and the ID token. Most of this information is highly sensitive, as is the image and its possible modifications. For this reason, the communication between them should be as secure as possible. The safest option for this communication is by encrypting the messages, although these encryption and decryption modules should be included in both platforms.

Due to the biometric process transferring from the conventional terminal to the new ID tokens, terminals in these new systems are relatively simple, and thus their cost is has been reduced significantly.

System management logic

Terminals are the point of interaction between the user and the global system. Because of this, the terminals should also be able to perform management tasks, such as possible changes or modifications to the tokens or the release of new keys, it is recommended that the system is also connected to central system for information, services or updating.

The connection to the central systems should be encrypted again or by means of dedicated lines to avoid possible attacks between the central system and the terminals, and as a result, to the tokens. Work within this area can be carried out on the logic management so as to include additional security countermeasures when updating firmware or securing the terminals from possible attacks, described in this dissertation as concealed attacks.

12.1.2.4 Multibiometrics

Thanks to the possibility that hardware offers regarding concurrent processes, a further possible ID token implementation is the development of multibiometric identification tasks. As has been mentioned in Chapter one, several multibiometric approaches can be followed. For the ID token proposed, a multi-algorithm system could be implemented to file improved performance results, and so only one sensor is required to acquire the data, and to perform both identification algorithms in the token at the same time.

The multibiometric token development using a hardware/software co-design can be carried out using the following structure described in this dissertation with respect to the ID token discussed in chapter 9

12.1.2.5 Combining biometric ID tokens with cryptography

The architecture proposed for our token can also be extended to further security applications. Of special interest within this area are those based on Biometrics and the additional security provided by cryptographic techniques, these techniques provide increased security levels by using an intelligent combination of hardware, ID tokens, Biometrics and cryptography.

Our interest is mainly focused on Biometric-key systems, as opposed to hash function systems, where these do not require the user to remember any password, but they still have the advantages of a password-based system (chapter 4). In iris technology, Hao [61] have presented the possibility of determining security codes from their iris codes, and thus opening an important research field.

These biometric-key systems combined with the ideas presented in this thesis may lead to the development of ID tokens based on Hardware/software co-design where the transformation between the feature vector and the key is carried out using a hardware module, as a result, no intruders can hinder communications or the process taking place, thereby reinforcing the security of the system. To achieve an optimum ID token architecture, the proposal presented in this thesis can by effectively used.

12.1.2.6 Reconfiguration

A commonly used tool in hardware designing, especially in FPGAs, is reconfiguration. This technology consists on giving the hardware area several uses during the process by means of the natural configuration capabilities they have. We have not carried out any reconfiguration in our proposal because, as has happened with software, possible intruders can try to access through this process of updating. The reconfiguration bitstream (new functions for the hardware area) is usually stored in an additional memory, which is read during the process, and the FPGA is able to reconfigure itself by complete occupancy of the bus and stopping the process during that time.

Although the reconfiguration is quite attractive in saving hardware area, it provides some other problems and therefore, the necessity of re-studying the hardware/software problem. If reconfiguration is desired, we should consider the space required for storing the bitstream as well as the time consumed while the FPGA is reconfiguring and is inactive for other processes, as well as the area required. The area now can change as in this process, some hardware is substituted by other, and therefore, the cost function should be considered in this issue. In finding the optimal solution, the problem now is an NP-hard problem and not NP-complete as the case we dealt with. Therefore, finding a solution is at least as hard as the hardest problems in NP, thus, requiring careful study.

Although some contributions have been made in saving time during the reconfiguration process [36], and reducing time, the problem related to security still can be found. Zeineddini proposed in [6] a solution for securing the reconfiguration bitstream by enciphering it using a cryptographic processor. However, this solution still requires more time than even a plain reconfiguration. Some further work combining these different perspectives should be made in order to obtain an effective ID token solution.

12.1.3 Progress expectations

In spite of possible drawbacks, a researcher must remain resolute on one solid proposal for future progress. This is the main conclusion that may wrap up our work during the development of this Thesis.

Base-group Further Research

This Thesis has prospered due to the efforts and cooperation of a work group determined to obtain the best results in the areas of research described in this Thesis. With each passing day, the knowledge acquired and the research carried out has led us to more ambitious quests regarding the specializations covered by the different members of our group.

In a professional sense, we feel compelled to continue further research in the areas of investigation of our proposal. We are convinced that our group has the experience, knowledge and will offer further contributions to this science within the fields being investigated.

Development Opportunities

The motivation, as demonstrated and mentioned, behind our proposals, which is backed up by research work carried out not only by our base-group colleagues but by all researchers within the field of identification, Iris Biometrics has been demonstrated to offer exciting new opportunities which are on the verge of future development.

Our research instinct has compelled us to put further effort into identifying such opportunities and has encouraged us to work in harmony with other researchers interested in contributing to optimum use of the iris for new biometric developments.

New horizons towards innovation

Researchers are but one of the many key elements required for innovation. This Thesis has provided a valid proposal that look towards new horizons and future applications of iris biometrics. It is now time to keep working towards innovative applications and further improvements to the conclusions presented here in this Thesis which offer realistic effective contributions to our community. Others must be willing to avail of our conclusions and take on the challenges that we have presented.

Appendix A

Hardware modules for Iris Biometrics

Due to paper limitations, Hardware modules for Iris Biometrics are not clearly visible in chapter 9. In this appendix, these figures are shown in bigger size to further examination.

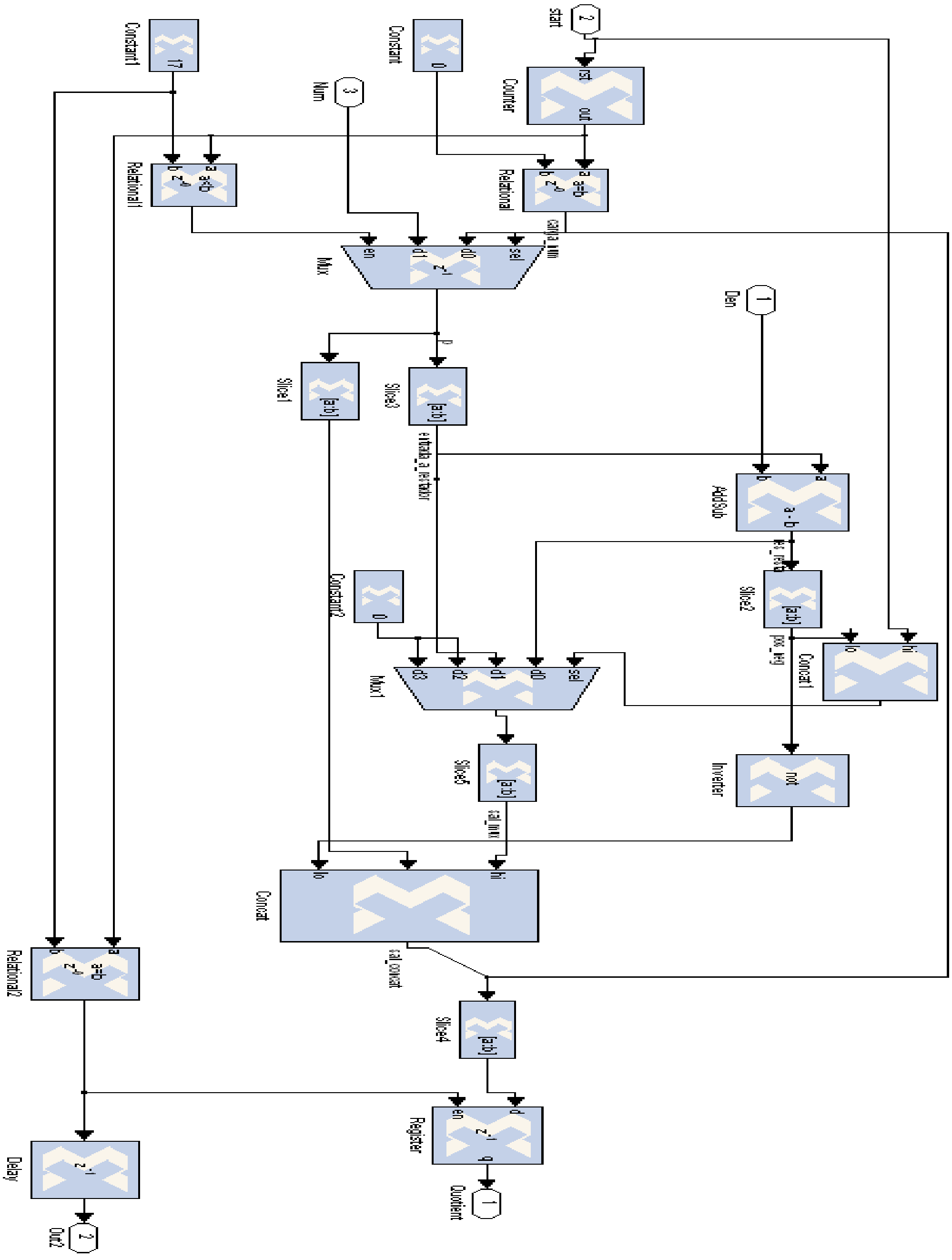


Figure A.1: Hardware implementation of the division operation

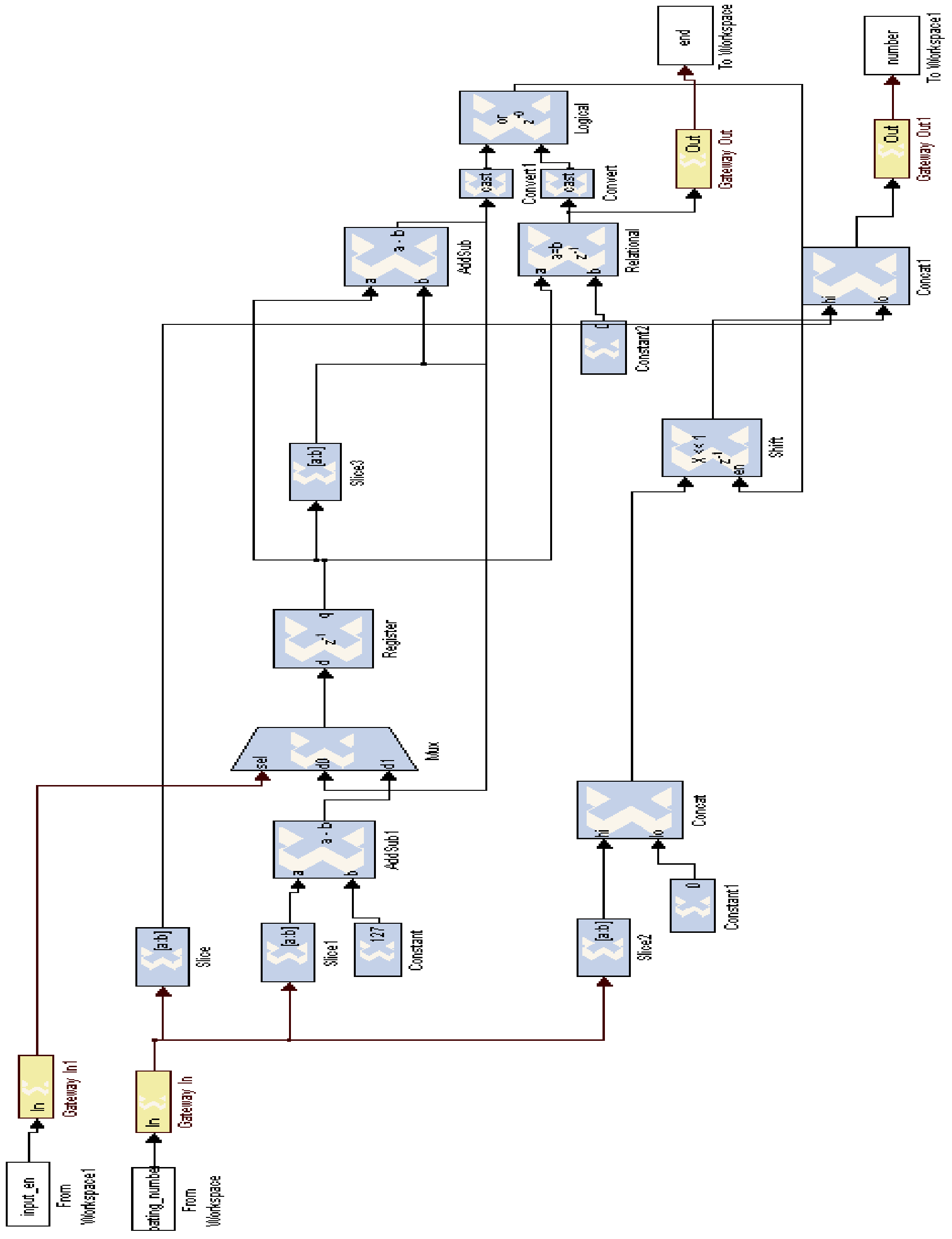


Figure A.2: Conversion data format module

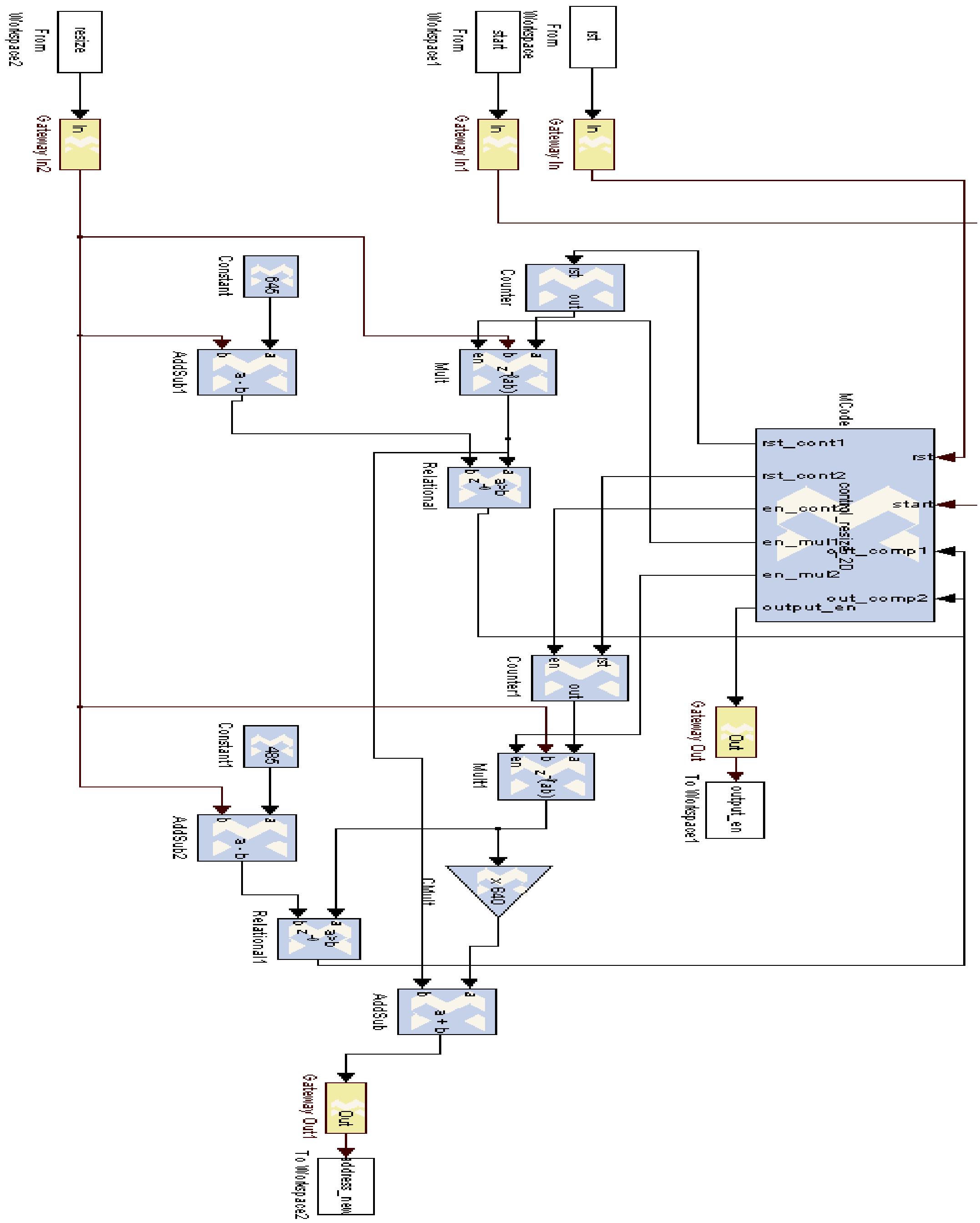
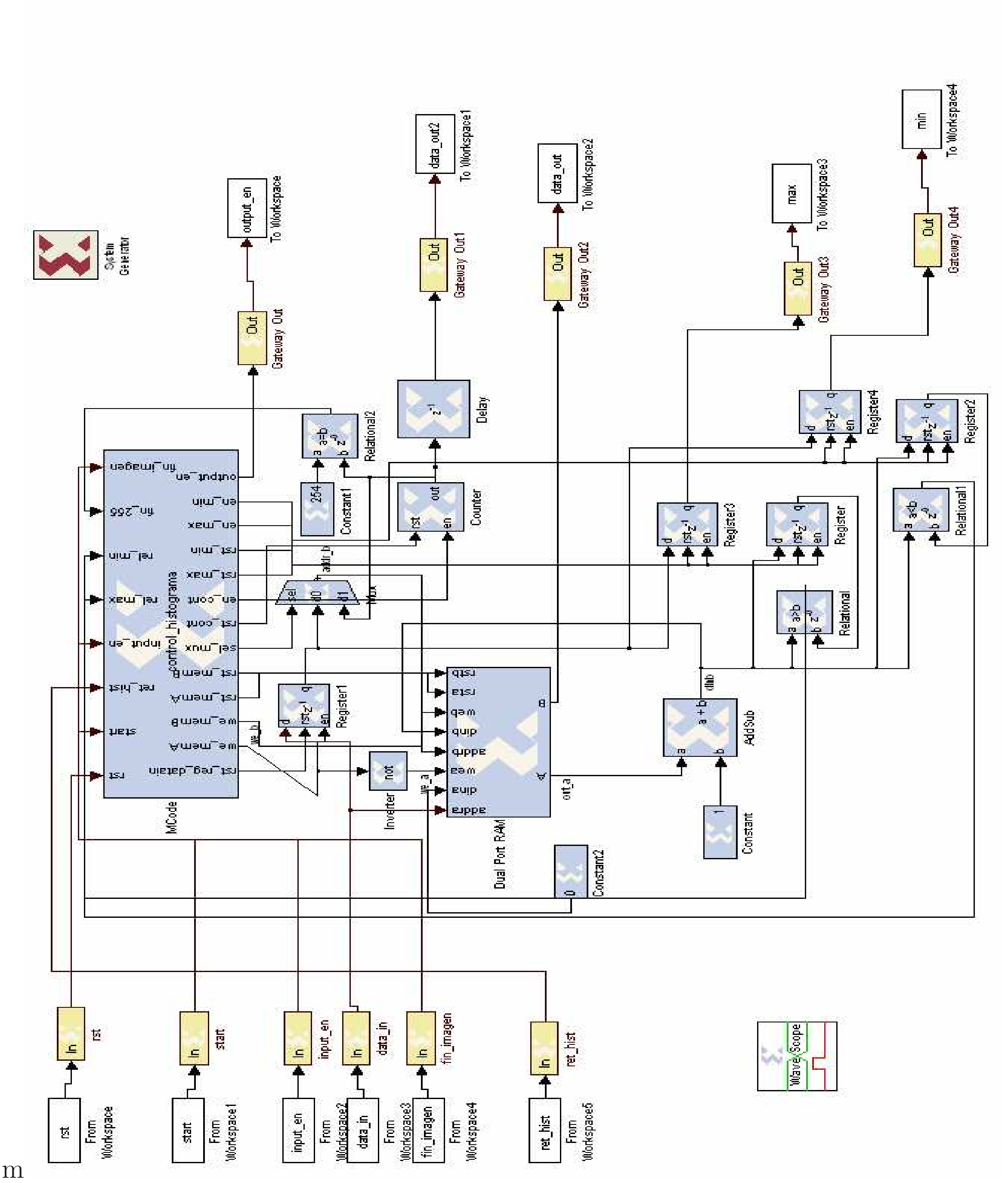


Figure A.3: Hardware implementation of the resize module



m

Figure A.4: Hardware histogram module

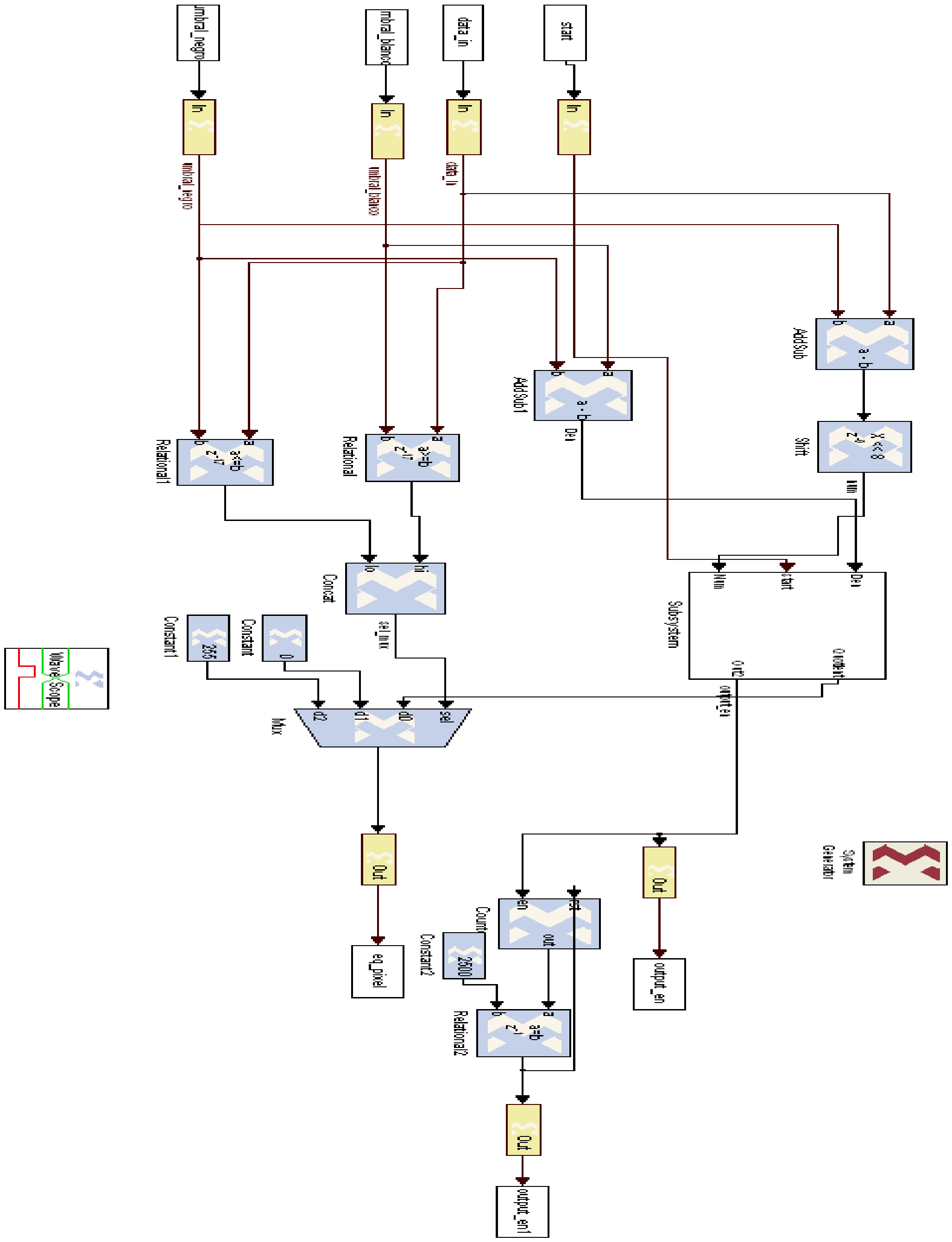
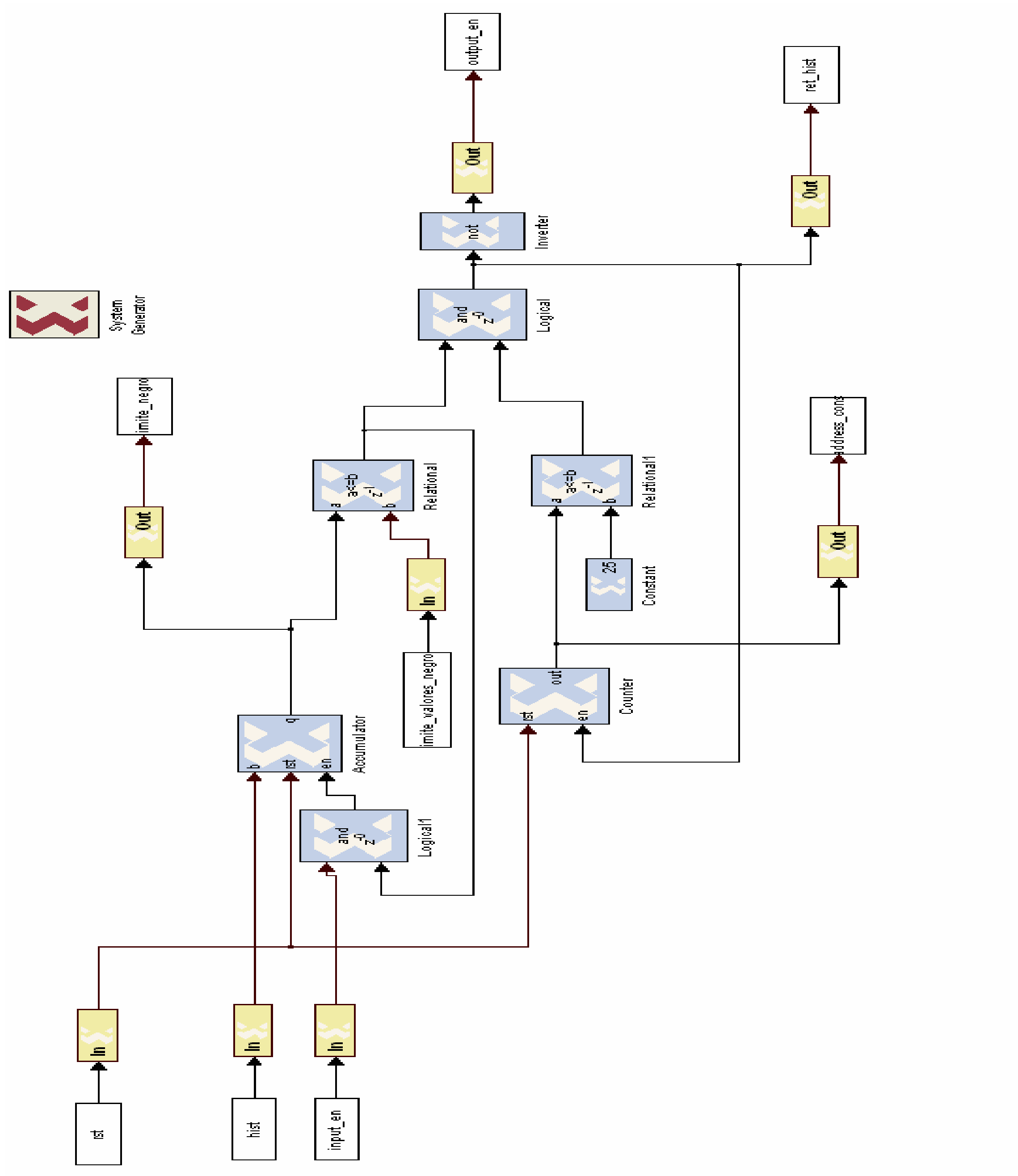


Figure A.5: Hardware equalizer module



m

Figure A.6: Black detection module

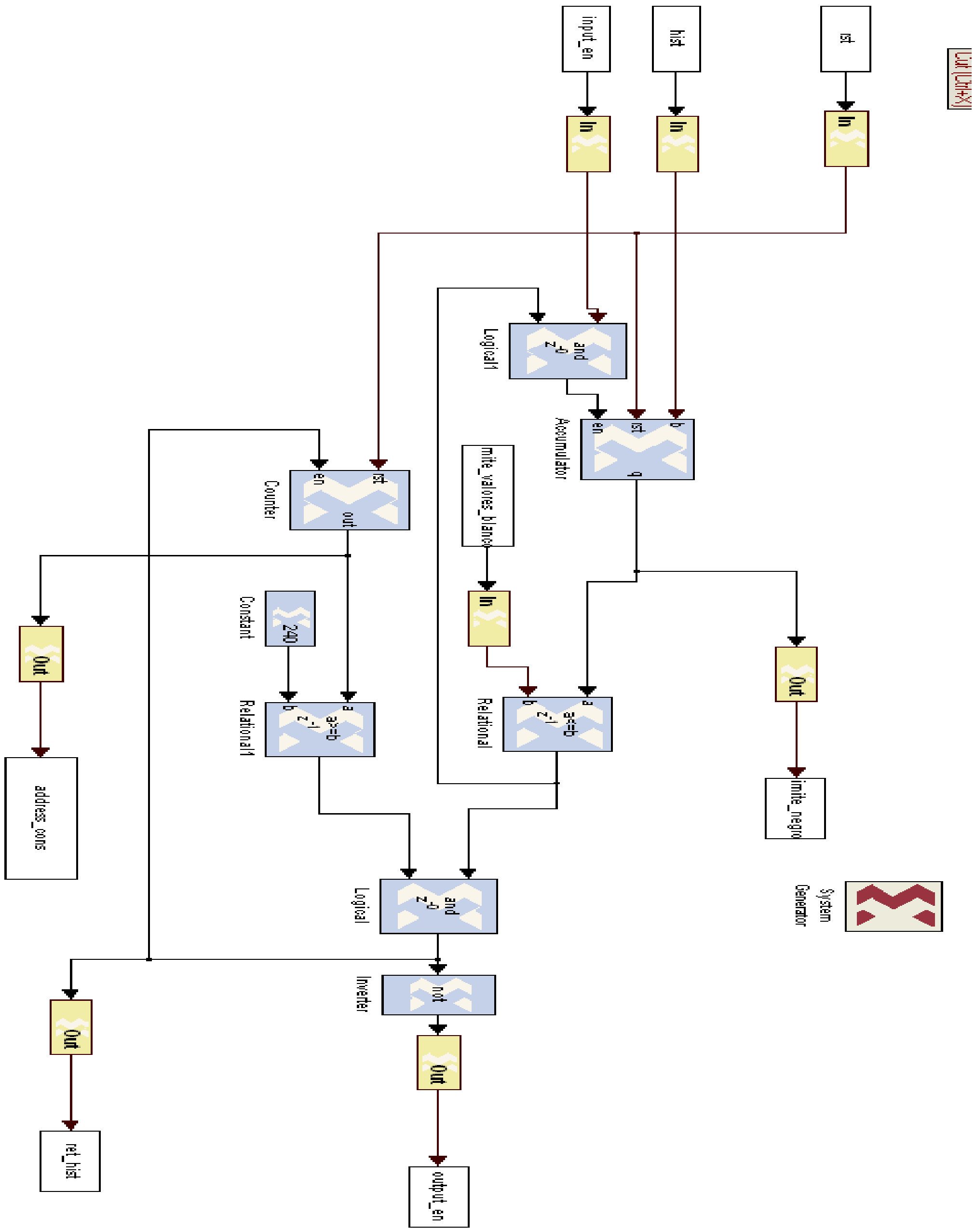


Figure A.7: White value detection module

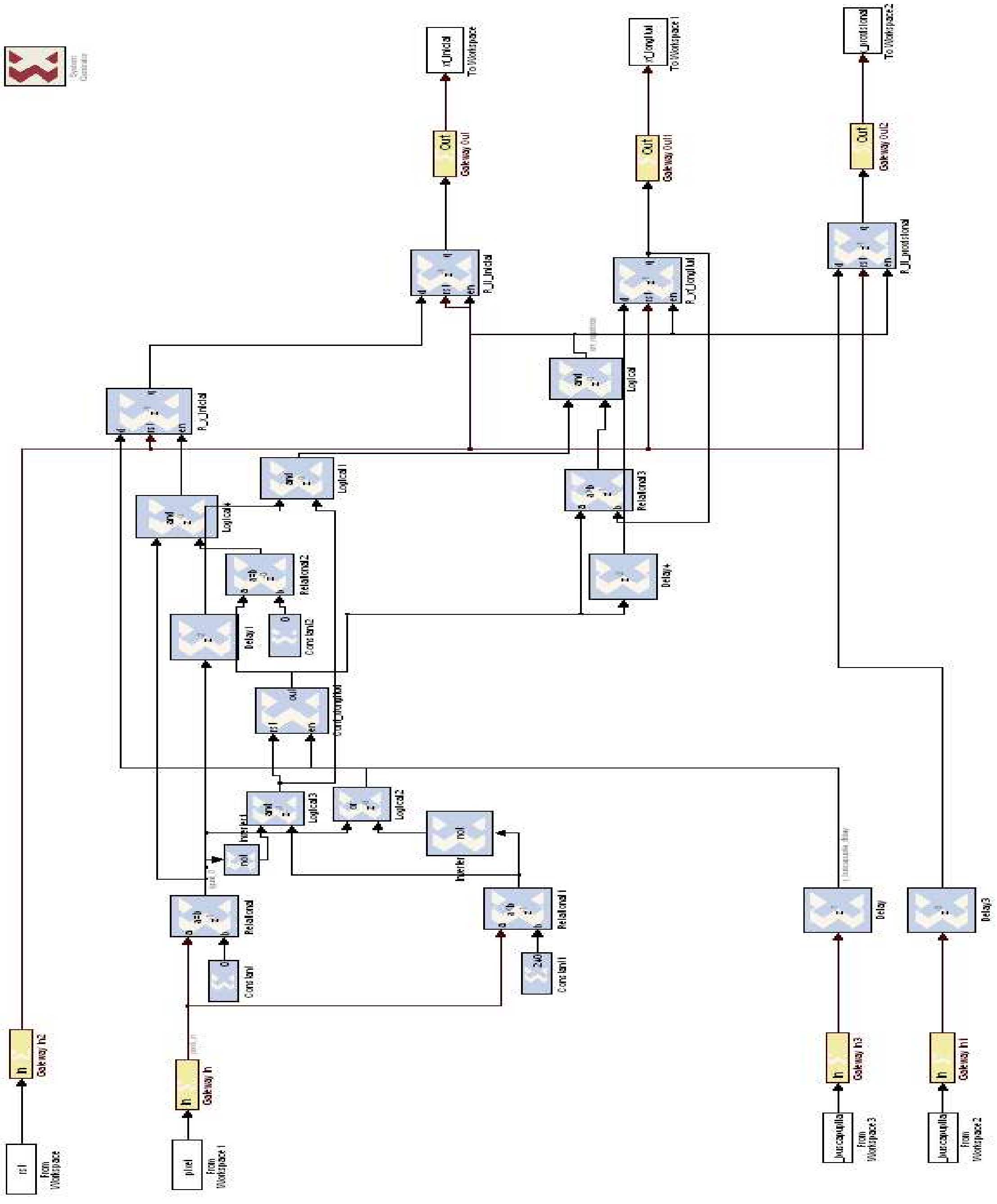


Figure A.8: Hardware module to detect black blocks

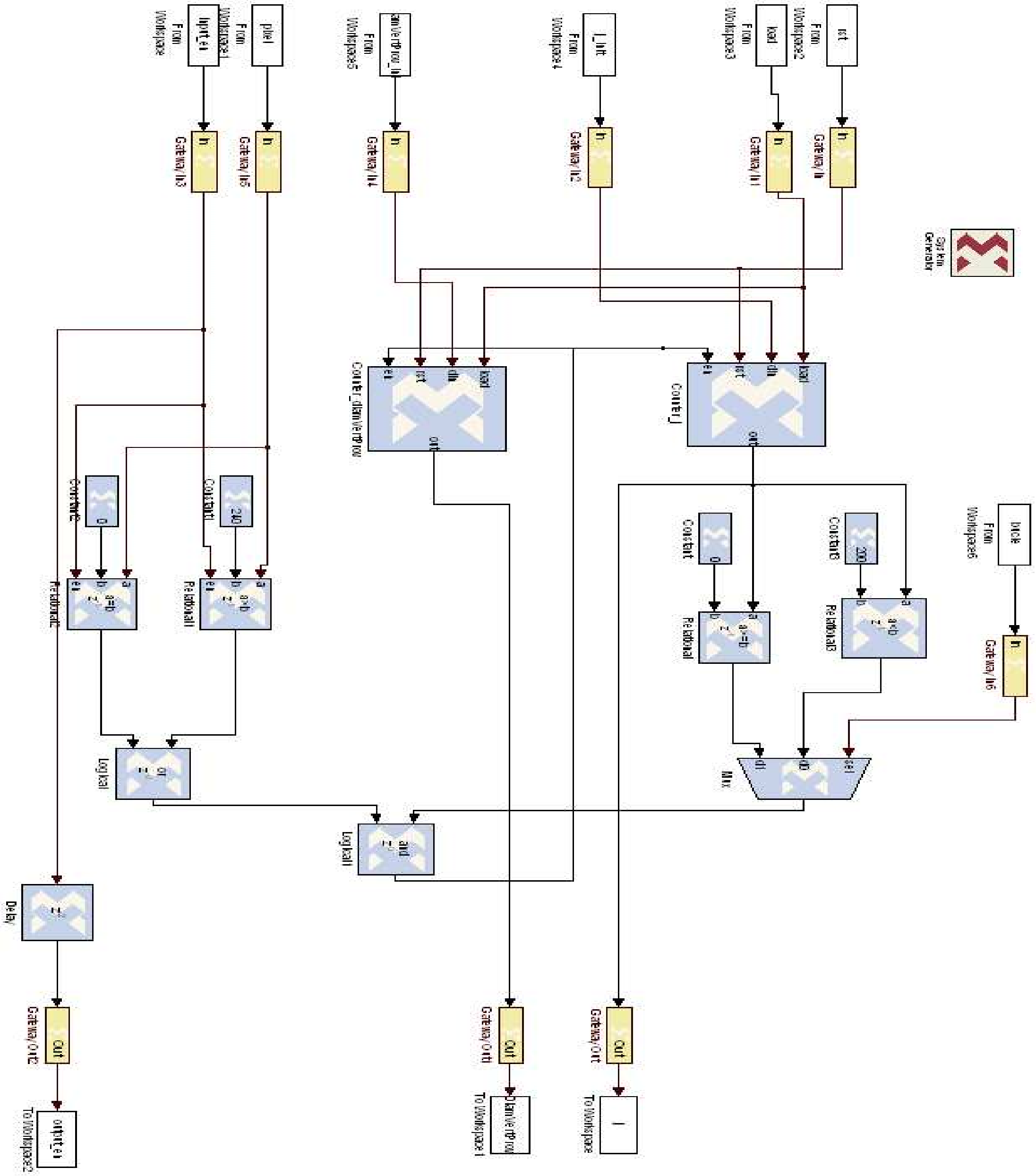


Figure A.9: Hardware module used to detect coarse pupil values

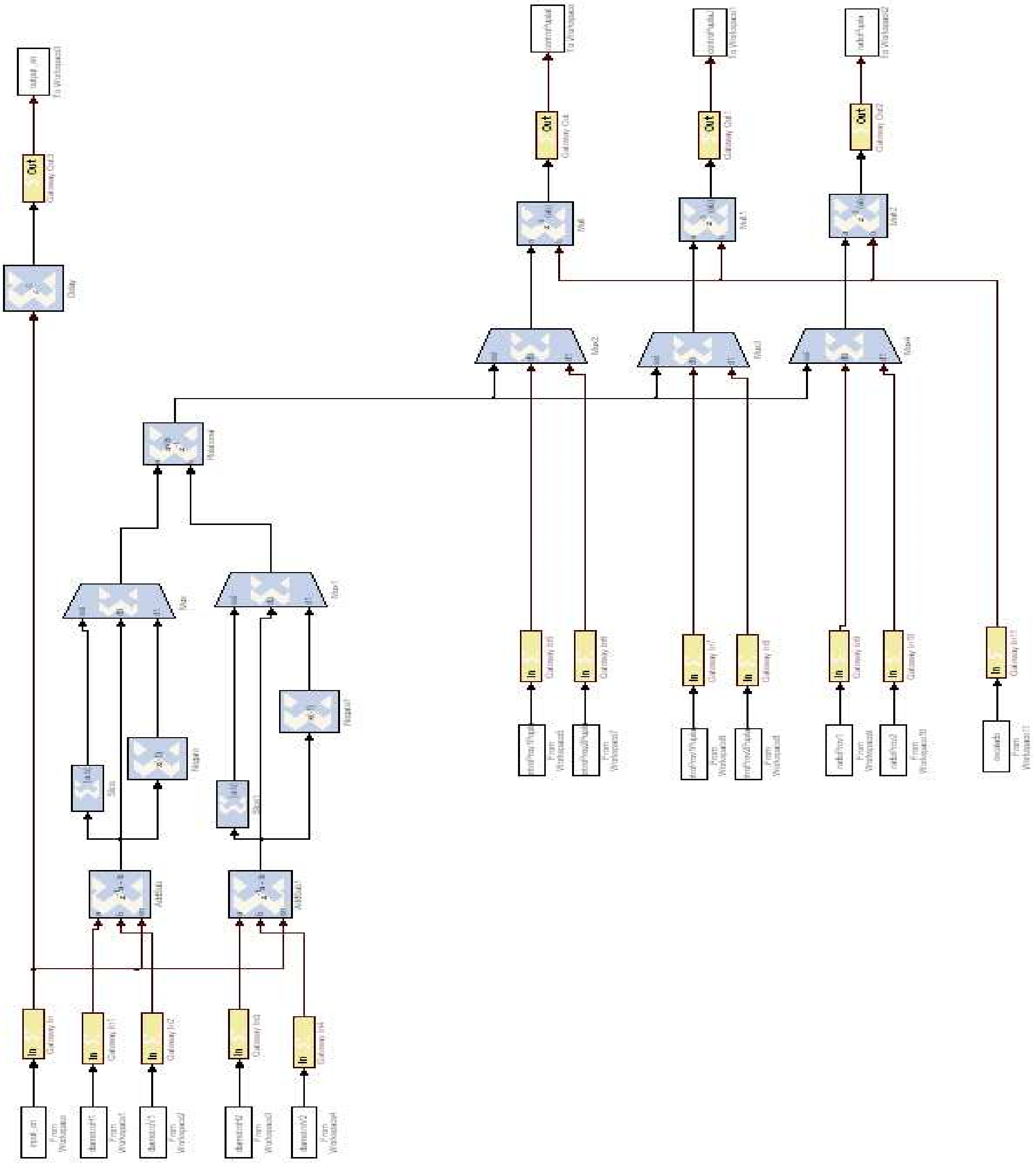


Figure A.10: Hardware module used to detect final coarse pupil values

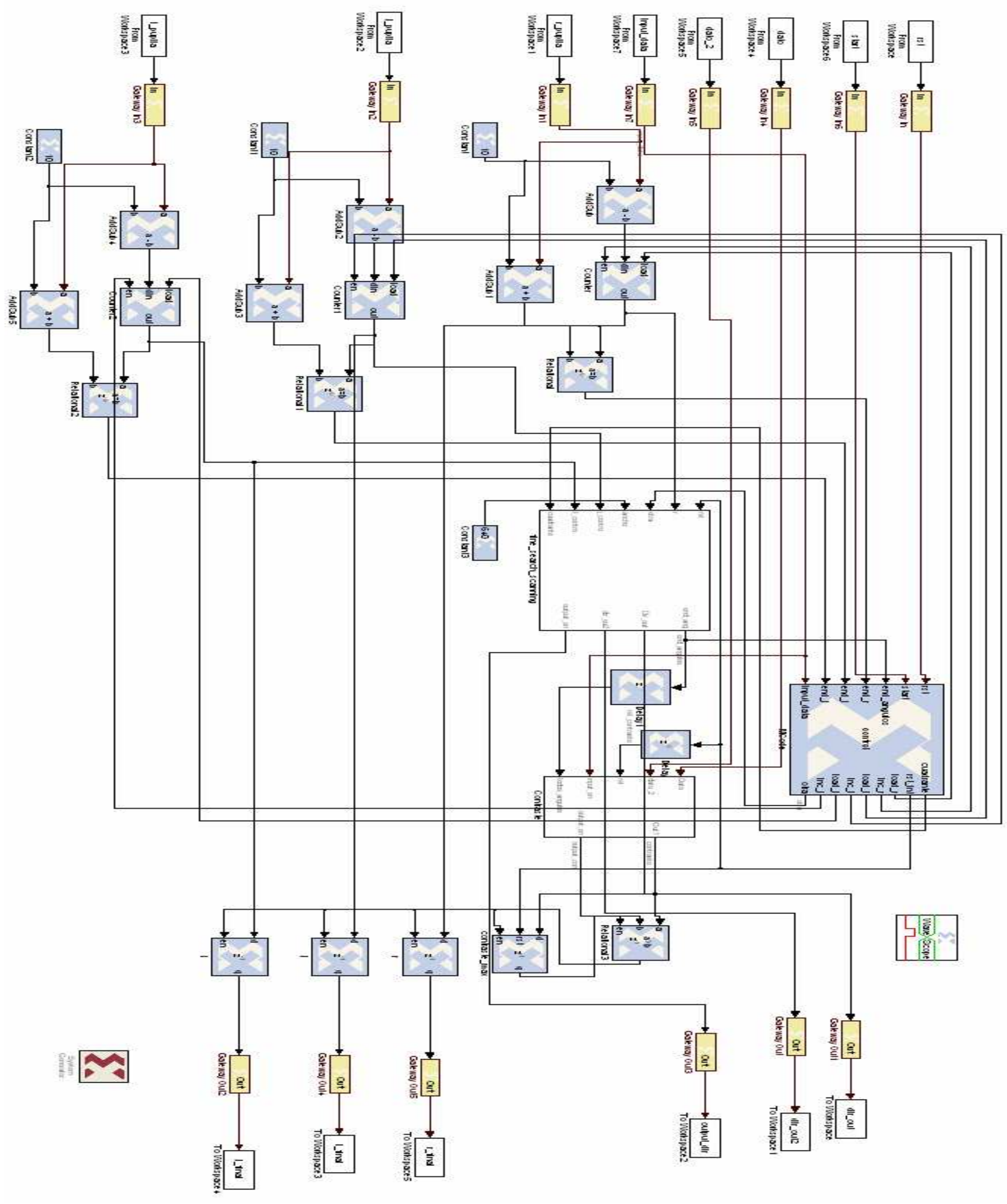
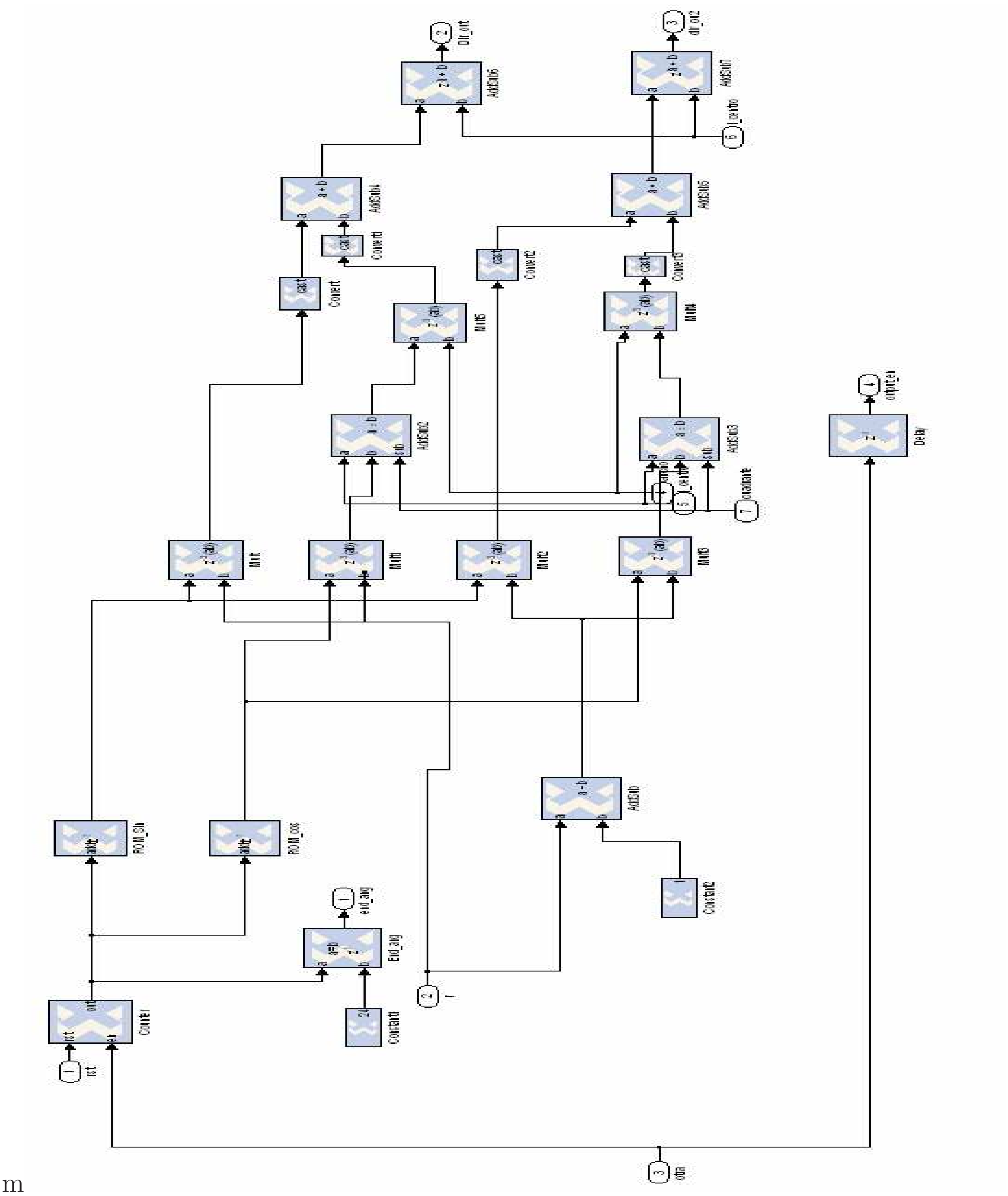


Figure A.11: Hardware module used to detect fine pupil values



m

Figure A.12: Hardware module used to search fine pupil values

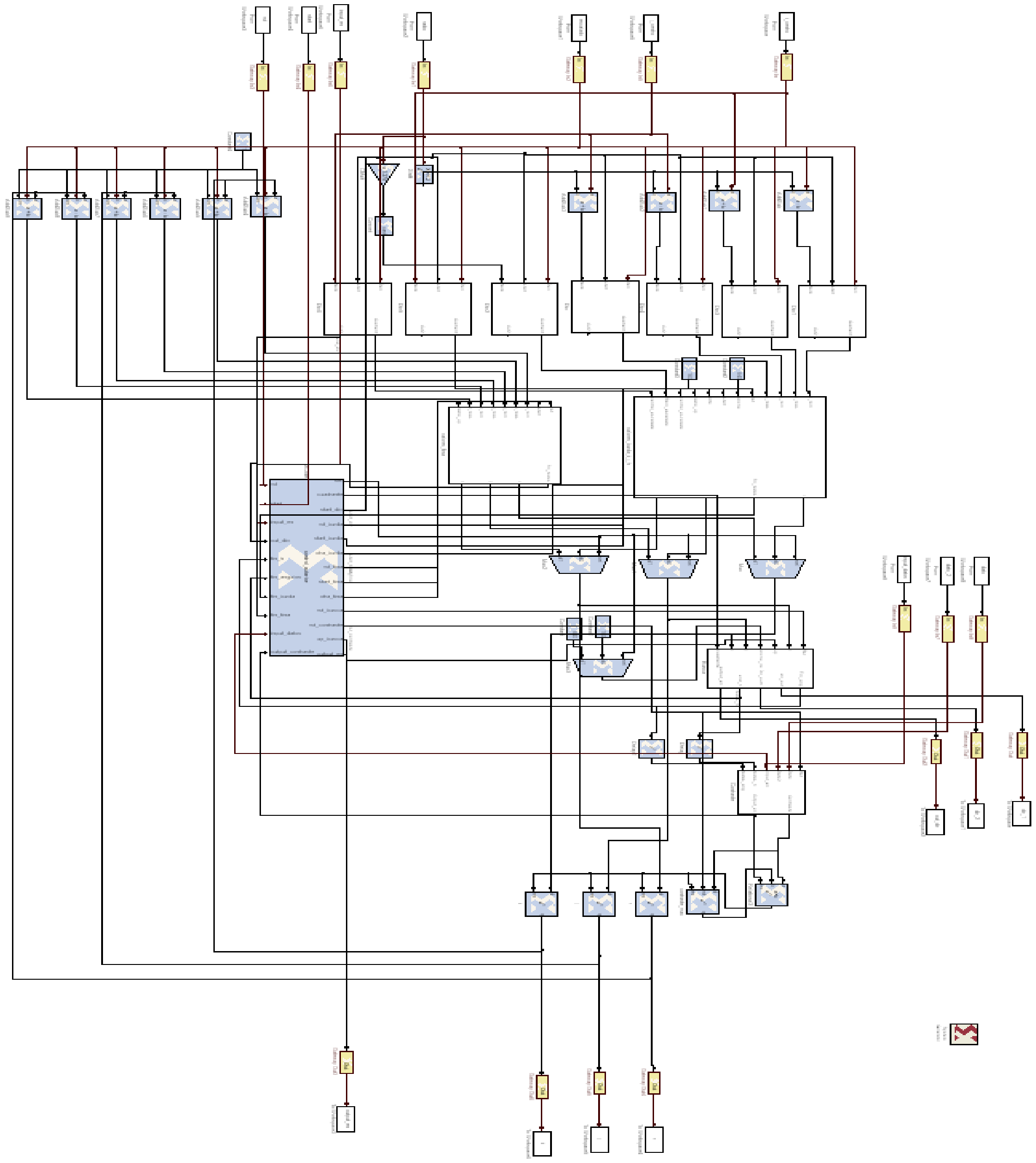
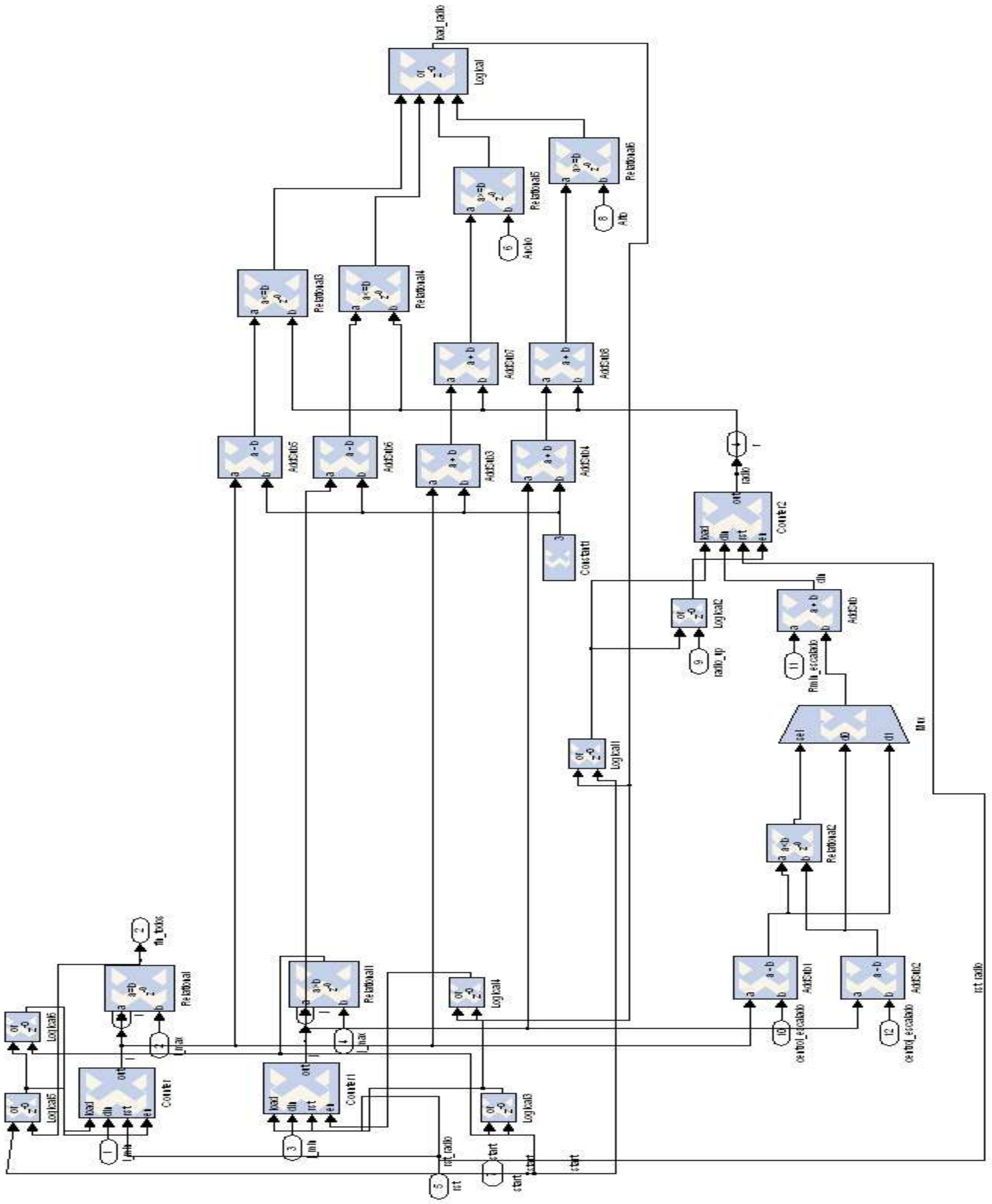


Figure A.13: Hardware module used to detect the outter iris boundary



m

Figure A.14: Hardware Coarse Search module used for the outer iris boundary

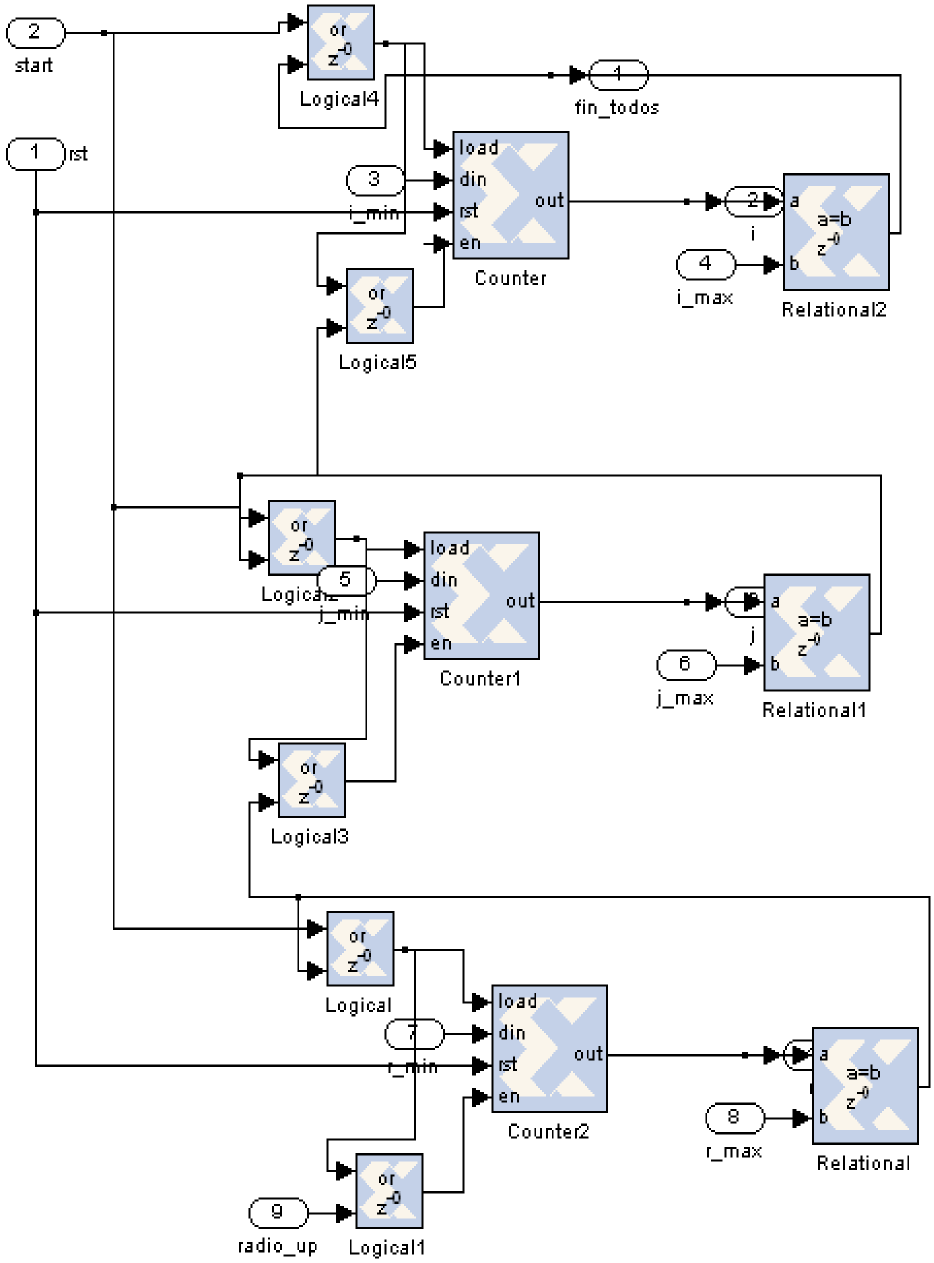


Figure A.15: Hardware Fine Search module used for the outer iris boundary

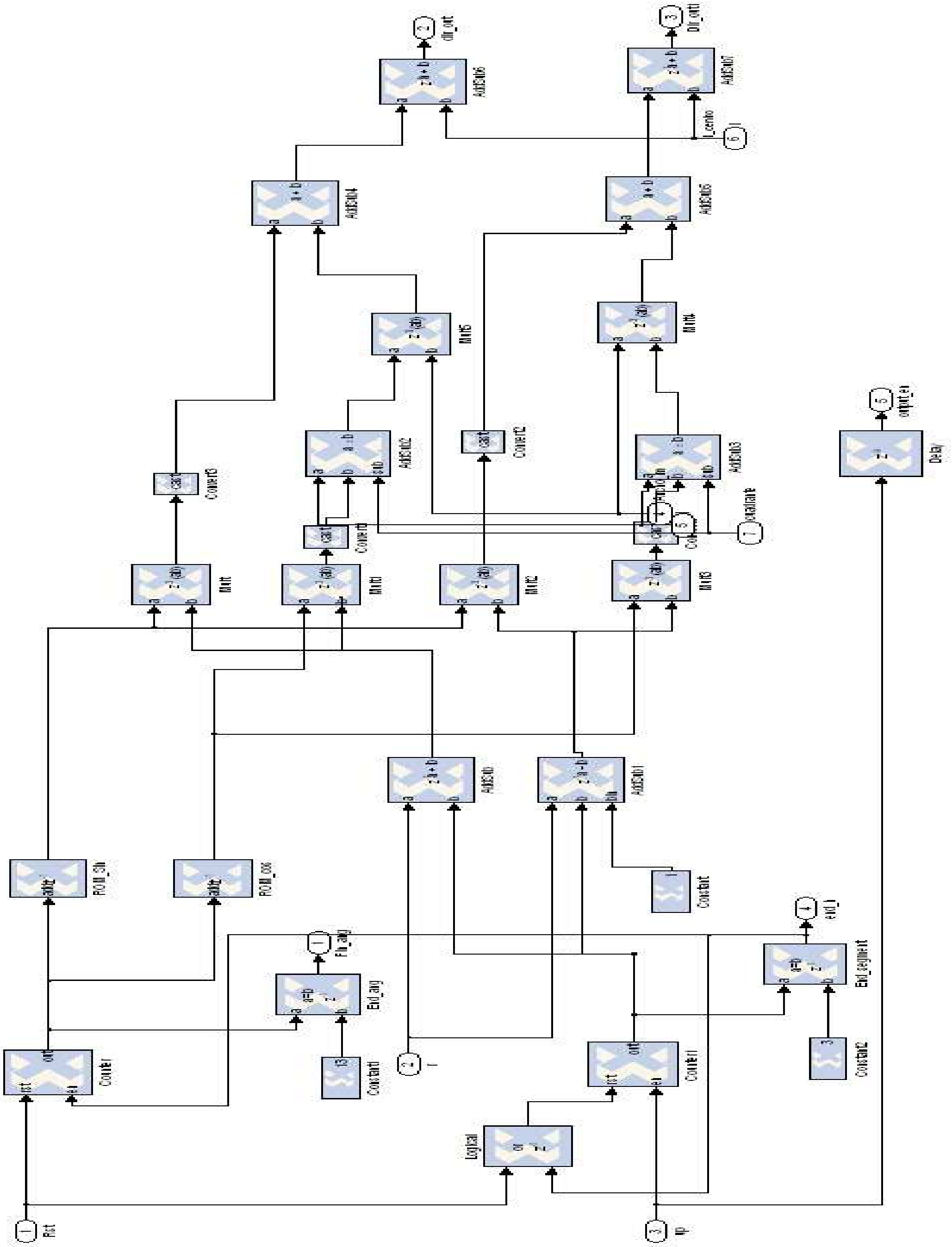


Figure A.16: Hardware search module used for the outer iris boundary

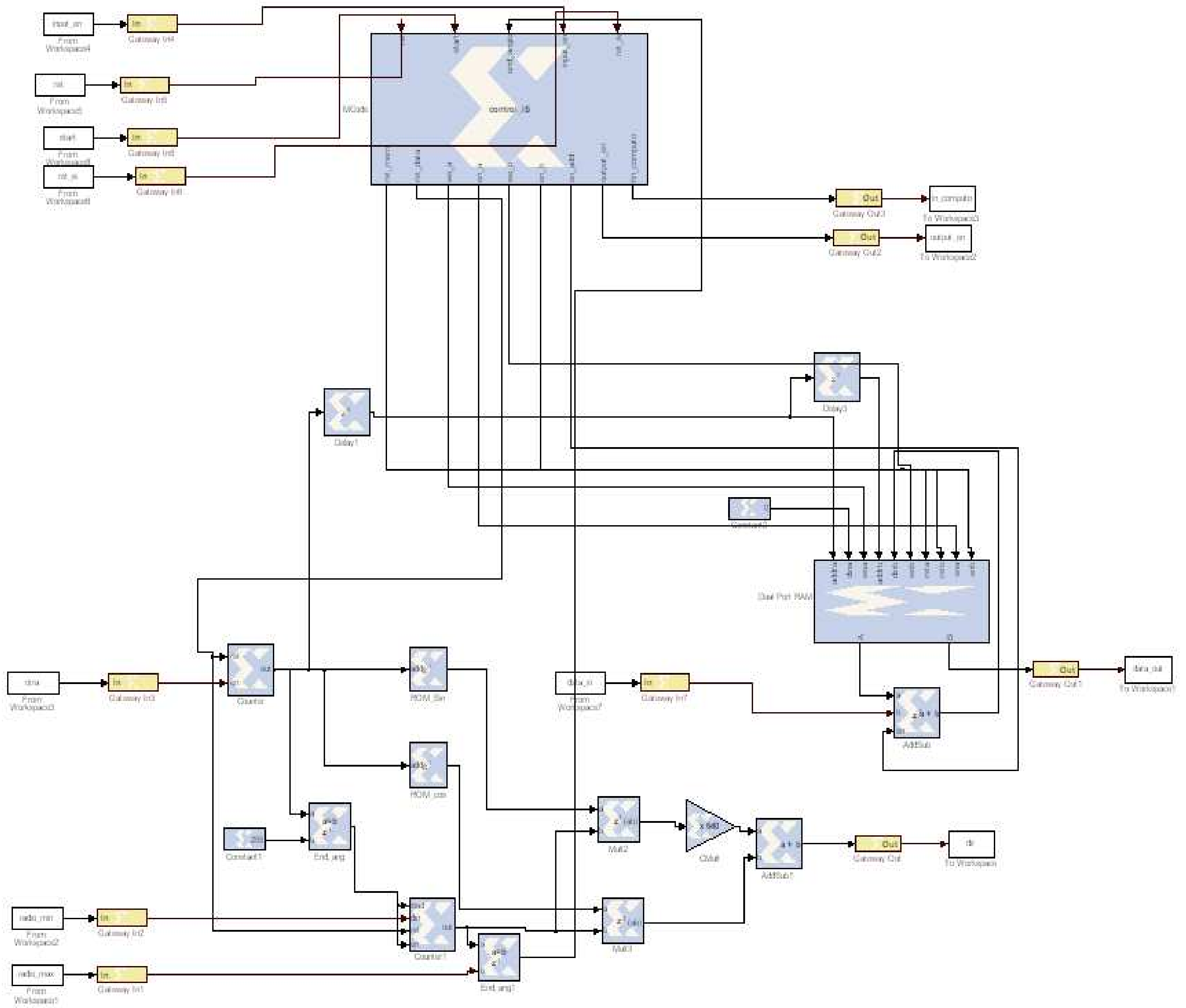


Figure A.17: Hardware implementation of the iris signature computation

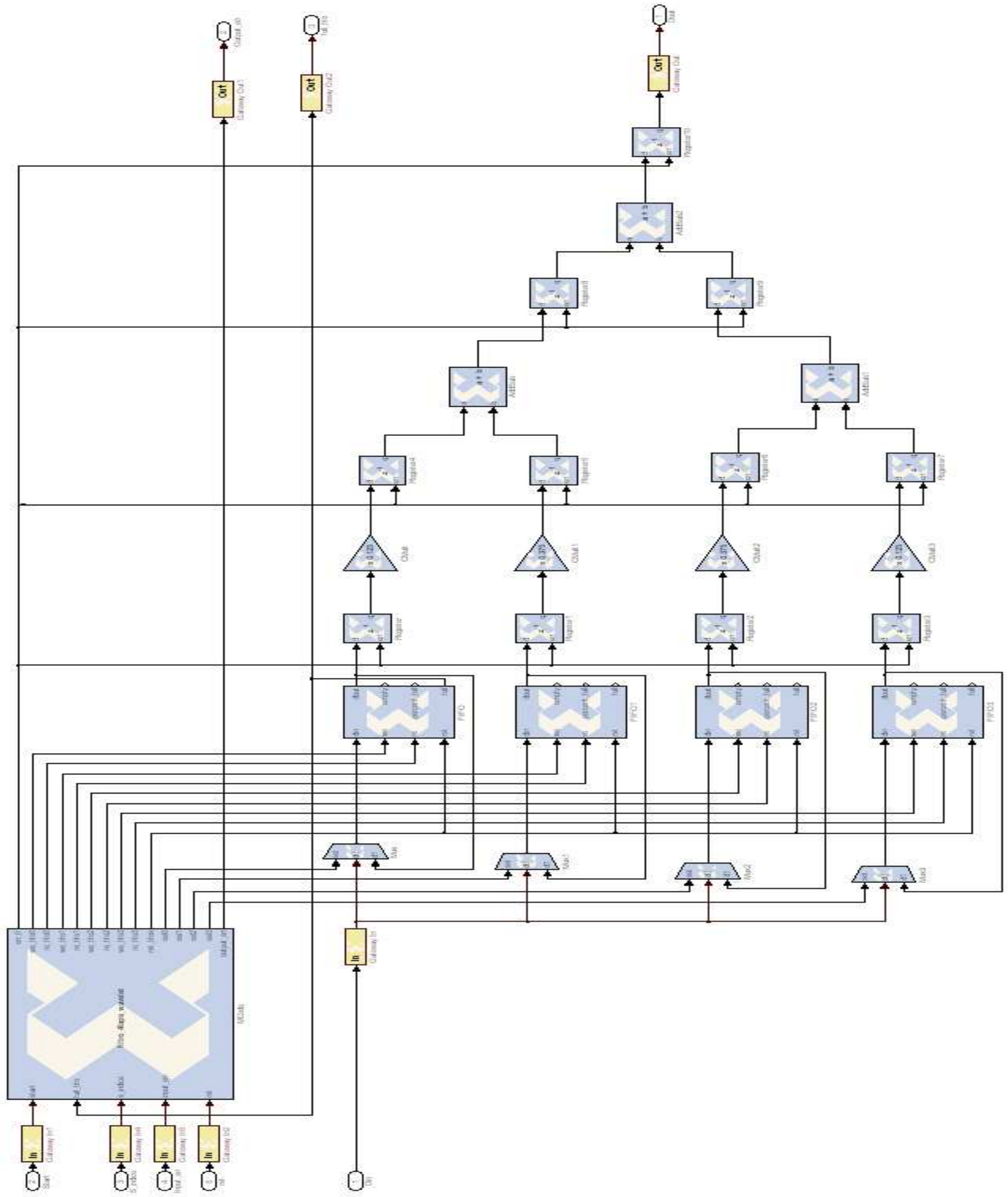


Figure A.18: Hardware implementation of the FWT high band filter

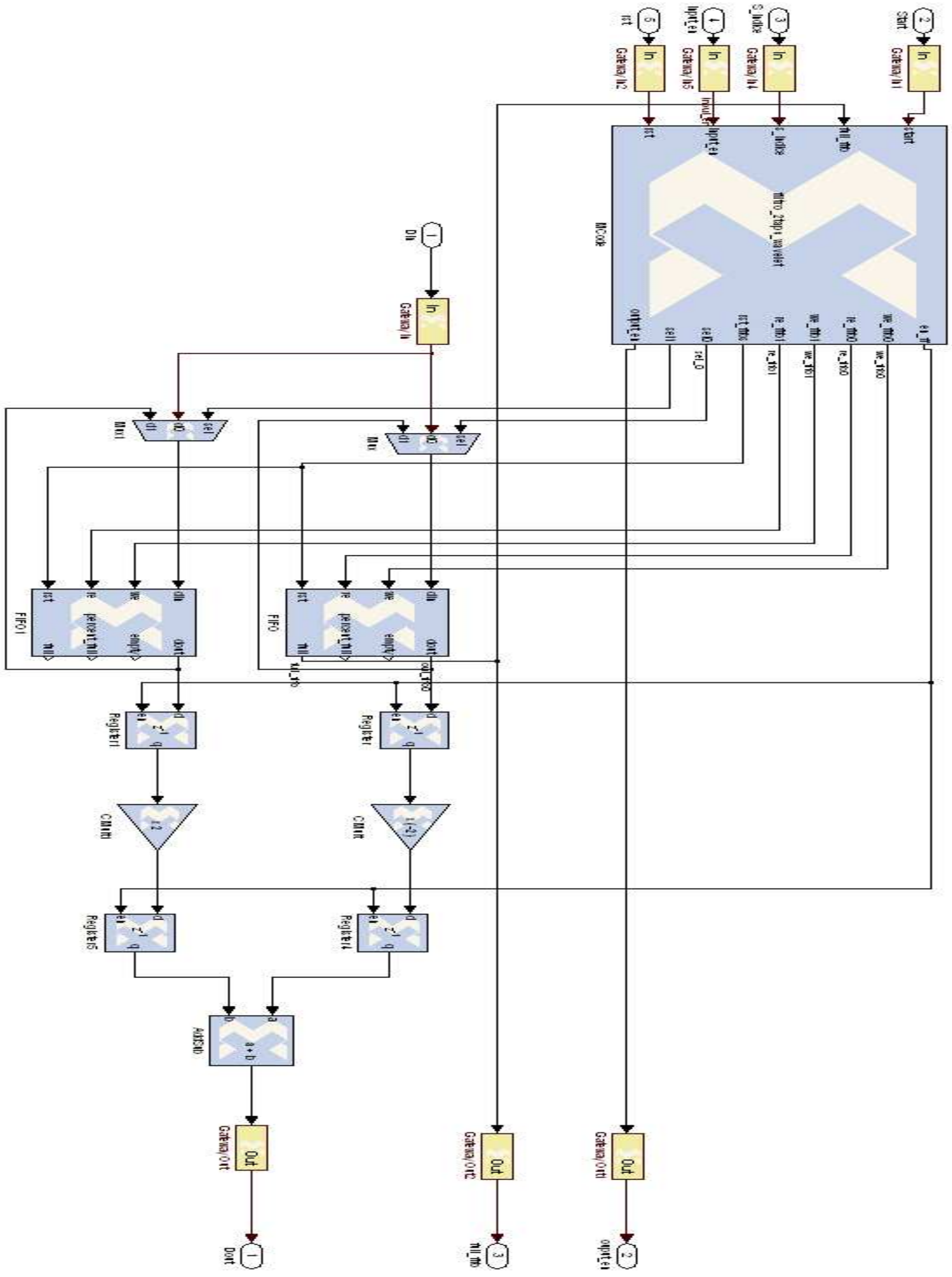


Figure A.19: Hardware implementation of the FWT low band filter

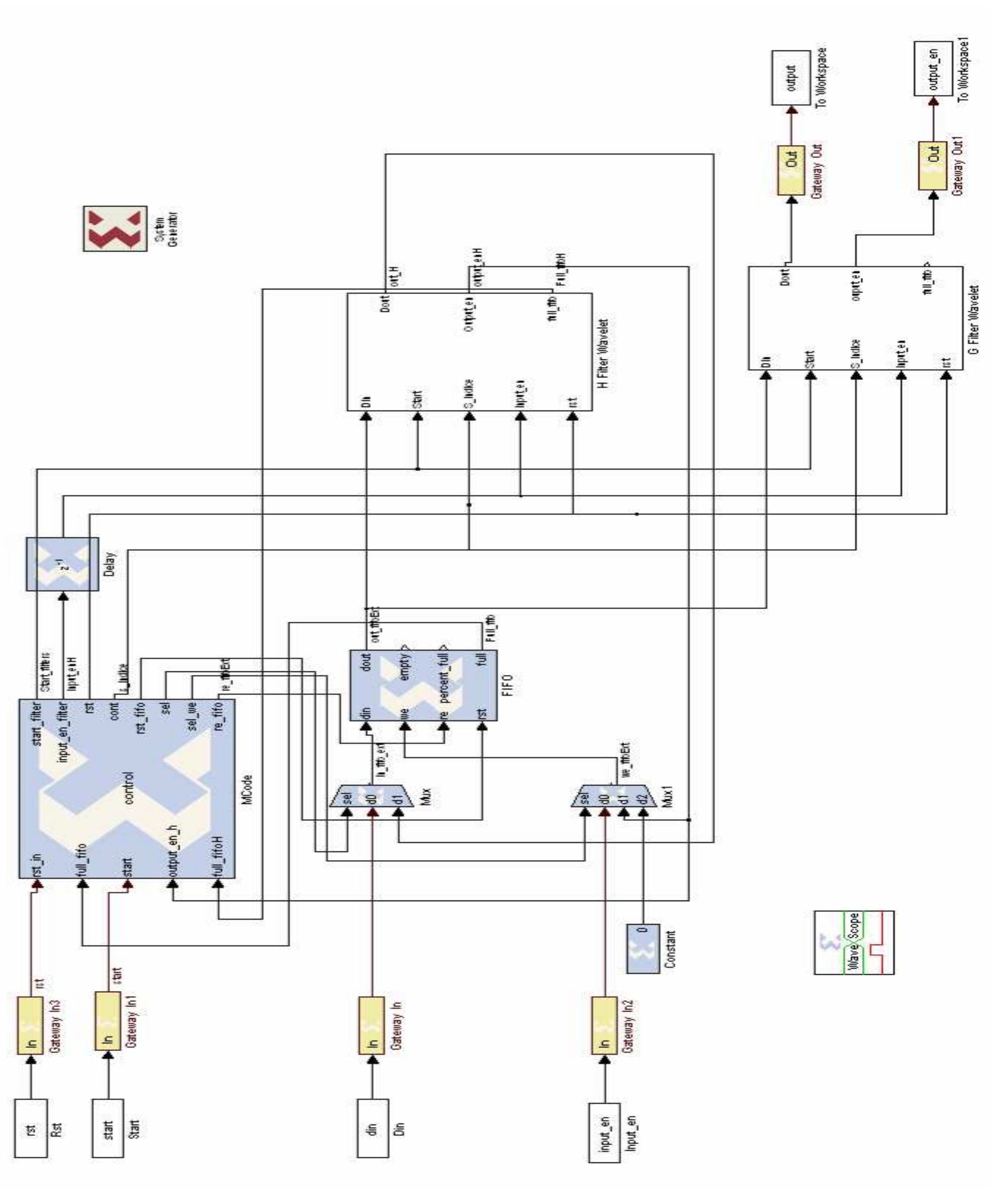


Figure A.20: Hardware implementation of the FWT

References

[1] ISO 19792. Iso/iec standard: Information technology – security techniques – security evaluation of biometrics, 2007. 58, 59, 62, 63

[2] B. Abidi, S. Huq, and M. Abidi. Fusion of visual, thermal, and range as a solution to illumination and pose restrictions in face recognition. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 325– 330, October 2004. 21

[3] Jarmo T. Alander. On optimal population size of genetic algorithms. *Proc of CompEuro'92: Computer Systems and Software Engineering*, May 1992. 95, 177

[4] Aphakia. http://en.wikipedia.org/wiki/aphakia, 2009. 34

[5] Péter Arató, Sándor Juhász, Zoltán Ádám Mann, András Orbán, and Dávid Papp. Hardware-software partitioning in embedded system design. *Proc. of the 2003 IEEE International Symposium on Intelligent Signal Processing*, September 2003. 86

[6] Zeineddini A.S. and Gaj K. Secure partial reconfiguration of fpgas. *Proc. of the IEEE International Conference on Field-Programmable Technology*, pages 155–162, December 2005. 244

[7] Jakob Axelsson. Hardware/software partitioning of real-time systems. *IEE Colloquium on partitioning in Hardware/software co-designs*, August 2002. 86

[8] K. Bae, S. Noh, and J. Kim. Iris feature extraction using independent component analysis. *Proc of International Conference on Audio and Video-Based Biometric Person Authentication*, pages 838–844, 2003. 46

[9] Belén Melián Batista and Fred Glover. Introducción a la búsqueda tabú. 94

[10] Arnold S. Berger. *Embedded systems Design: An Introduction to Processes, Tools & Techniques*. Ed. CMP Books, 2001. 73

[11] Sudarshan Bernejee, Elaheh Bozorgzadeh, and Nikil D. Dutt. Integrating physical constraints in hw-sw partitioning for architectures with partial dynamic reconfiguration. *IEEE Transactions on very large scale integration (VLSI) systems*, **14**[11]:1189–1202, November 2006. 91

[12] Biometrix. http://www.biometrix.at/. 201

[13] W. W. Boles and B.Boashash. A human identification technique unisg images of the iris and wavelet transform. *IEEE Transactions on Signal Processing*, **46**[4]:1185–1188, April 1998. 46, 47

[14] Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, and Andrew W. Senior. *Guide to Biometrics*. Springer, 2004. 7, 8, 11, 13, 16, 19, 21, 25

[15] Bill Burr. Biometric and electronic authentication. In *Biometrics Consortium Conference, 2005*. Biometrics Consortium Conference, Available on line: http://www.biometrics.org/bc2005, September 2005. 61, 104

[16] William E. Burr, Donna F. Dodson, and W. Timolthy Polk. Electronic authentication guideline – information security. Technical report, Nist National Institute of Standards and Technology, 2006. 61, 109

[17] IBM On chip Peripheral Bus Specification. http://ens.ewi.tudelft.nl/education/courses/et4351/opb_ibm_spec.pdf. 213, 214

[18] Carlos A. Coello Coello. Búsqueda tabú: Evitando lo prohibido. 94

[19] O. Coltell, J.M. Badfa, and G. Torres. Biometric identification system based on keyboard filtering. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 203–209, October 1999. 26

[20] Fingerprint Verification Competition. http://bias.csr.unibo.it/fvc2006/, 2006. 17

[21] AES IP Core. http://www.opencores.org/project,aes_core. 140

[22] J. Daugman. How iris recognition works 2d focus assessment at the video frame rate appendix. *IEEE transactions on Circuits and Systems for Video Technology*, **14**[1], January 2004. 237

[23] John Daugman. http://www.cl.cam.ac.uk/ jgd1000, 2003. 33, 34, 39

[24] John Daugman. Interview with dr. john daugman, cambridge university. *findBiometrics.com*, december 2004. 241

[25] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparison. *Proceeding of IEEE*, **94**[11]:1927–1935, November 2006. xxv, 31, 47, 202, 203, 208, 218, 219, 223

[26] John Daugman. New methods in iris recognition. *IEEE Transaction on Systems, Man and Cybernetics - Part B: Cybernetics*, **37**[5]:1167–1175, October 2007. 44, 226

[27] John Daugman and Cathryn Downing. Effect of severe image compression on iris recognition performance. *IEEE Transaction on Information Forensics and Security*, **3**[1]:52–61, March 2008. 48, 56

[28] John G. Daugman. High confidence visual recognition of persons by a test of statistical independece. *IEEE TRansaction on Pattern Analysis and Machine Intelligence*, **15**[11]:1148–1161, November 1993. 33, 42, 44, 45, 47, 203

[29] Dirección General de la Policía y la Guardia Civil Española. Documento nacional de identidad español: http://www.dnielectronico.es, 2005. 10

[30] Giovanni de Micheli and Rajesh K. Gupta. Hardware/software co-design. *Proceedings of the IEEE*, **85**[3]:349–365, March 1997. 80, 84, 86, 89

[31] M. Rafael Diaz, C.M. Travieso, J.B. Alonso, and M.A. Ferrer. Biometric system based in the feature of hand palm. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 136– 139, October 2004. 22

[32] Computer Security Division. Nist sp 800-63: Technical authentication framework for remote e-authentication. Technical report, NIST: National Institution of Standarts and Technology, 2006. 103, 104

[33] R. Dömer, D. Gajski, and J. Zhu. Specification and design of embedded systems. *IT+TI Magazine*, [3], 1998. xxiv, 85

[34] Czech Republic Dpt. of Computer Science, Palacky University in Olomouc. Upol database: http://phoenix.inf.upol.cz/iris/, 2004. 41

[35] Portugal Dpt. of Computer Science, University of Beira Interior. Ubiris database: http://iris.di.ubi.pt, 2004. 41

[36] Canto E., F. Fons, and Lopez M. Self-recofigurable embedded systems on spartan-3. *Proc. of the IEEE International Conference on Field-Programmable and Applications*, 2008. 244

[37] Thomas Ea, Frédéric Amiel, Alicja Michalowska, Florence Rossant, and Amara Amara. Erosion and dilatation implementation for iris recognition system using different techniques on sopc. *Proc. of the XXI International Conference on Desgin of Circuits and Integrated Systems*, 2006. 87, 117

[38] Petru Eles, Zebo Peng, Krzysztof Kuchcinski, and Alexa Doboli. Hardware/software partitioning with iterative improvements heuristics. *Proc. of the 9th International Symposium on System Synthesis*, November 1996. 89

[39] Kukula E.P., Elliott, S.J., B.P. Gresock, and N.W. Dunning. Defining habituation using hand geometry. *Proc. of the 2007 IEEE Workshop on Automatic Identification Advanced Technologies*, pages 242–246, June 2007. 22

[40] Rolf Ernst. Codesign of embedded systems: Status and trends. *IEEE Design & Test of Computers*, **15**[2]:45–54, April-June 1998. 80, 84

[41] NIST Iris Challenge Evaluation. http://www.fda.gov/ohrms/dockets/dockets/00d1538/00d-1538-mm00025-02.pdf, 2006. 17

[42] Marcos Faundez-Zanuy. Biometric security technology. *IEEE A & E Systems Magazine*, **21**[6]:15–26, June 2006. 8, 18

[43] Marcos Faúndez-Zanuy, Martin Hagmüller, and Gernot Kubin. Speaker identification security improvement by means of speech watermarking. *Pattern Recognition*, **40**[11]:3027–3034, 2007. 67

[44] Belen Fernandez-Saavedra, Raul Sanchez-Reillo, and Raul Alonso-Moreno. Evaluation methodology based on cem for testing environmental influence in biometric devices. *Proc. of the 9th International Conference on Common Criteria*, 2008. 60

[45] Belen Fernandez-Saavedra, Raul Sanchez-Reillo, Raul Alonso-Moreno, and R. Mueller. Evaluation methodology for analyzing environment influence in biometrics. *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008)*, 2008. 60

[46] M.A. Ferrer, A. Morales, C.M. Travieso, and J.B. Alonso. Low cost multimodal biometric identification system based on hand geometry, palm and finger print texture. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 52–58, October 2007. 22

[47] M. fons, F. Fons, and E. Canto. Design of an embedded fingerprint matcher system. *Proc Of 10th IEEE International Symposium on Consumer Electronics ISCE'06*, pages 1–6, 2006. 87, 199

[48] M. fons, F. Fons, and E. Canto. Hardware coprocesor design of a fingerprint aligment processor. *Proc Of 14th International Conference on Mixed Design of Integrated Circuits and Systems MIXDES'07*, pages 661–666, 2007. 87, 117, 199

[49] M. Fons, F. Fons, E. Canto, and M. Lopez. Hardware-software co-design of a fingerprint matcher on card. *Proc Of 50th IEEE International Conference on Electro/information Technology*, pages 113–118, 2006. 87, 199

[50] Center for Biometrics and Security Research. Casia database: http://www.cbsr.ia.ac.cn/irisdatabase.htm, 2008. 40, 46

[51] Statistical Demands for Identification vs Verification. http://www.cl.cam.ac.uk/ jgd1000/veri/veri.html, 2005. 14

[52] International Organization for Standardization. Standard iso 19794:information technology – biometric data interchange formats – part 6: Iris image data, 2005. 54

[53] J. Galbally-Herrero, J. Fierrez-Aguilar, J. D. Rodriguez-Gonzalez, F. Alonso-Fernandez, Javier Ortega-Garcia, and M. Tapiador. On the vulnerability of fingerprint verification systems to fake

fingerprints attacks. *Proc. of the 40th IEEE International Carnahan Conferences Security Technology*, pages 130–136, October 2006. 62

[54] Michel Gendreau. An introduction to tabu search. 94

[55] Fred Glover and Belén Melián. Tabu search. *Revista iberoamericana de inteligencia artificial*, [19]:29–45, 2003. 94, 134

[56] K. Grabowski, W. Sankowski, M. Zubert, and M. Napieralska. Focus assessment issues in iris image acquisition system. *Proc. of the 14th International Conference of Mixed Design*, pages 628–631, June 2007. 38

[57] Biometric Group. http://www.biometricgroup.com/, 2009. xxiii, 20, 28

[58] Int. Biometrics Group. Independent testing of iris recognition technology. Technical report, U.S. Dept. Homeland Security, 2005. 38

[59] Javier Macías Guarasa. Introducción al simulated annealing. 93

[60] Rajesh K. Gupta. Embedded processors. Project Report for ICS 212, March 2000. 75, 77, 89

[61] Feng Hao, Ross Anderson, and John Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, **55**[9]:1081–1088, September 2006. 47, 68, 243

[62] David L. Harris, Stuart F. Oberman, and Mark A. Horowitz. Srt division architectures and implementations. *Proc. of the 13th IEEE Symposium on Computer Arithmetic*, pages 18–25, 1997. 205

[63] Yuqing He, Jiali Cui, Tieniu Tan, and Yangsheng Wang. Key techniques and methods for imaging iris in focus. *Proc. 18th Internation Conference on Pattern Recognition*, **4**:557–561, August 2006. 38

[64] Javier Hormigo, Manuel Sanchez, Mario A. Gonzalez, Gerardo Bandera, and Julio Villalba. Optimized fpga implementation of trigonometric functions with large input argument. *Proc. International Conference of Design of Circuits and Integrated Systems*, pages 252–255, 2004. 166

[65] http://csrc.nist.gov/archive/aes/index.html. Advanced encryption standard. 140

[66] HTTP://OCW.KFUPM.EDU.SA/USER/EE55601/07 TS-MODIFIED.DOC. Tabu search algorithm. xxiv, 96

[67] HTTP://WWW.AI JUNKIE.COM/GA/IONTRO/GAT2.HTML. The genetic algorithm. 95, 135

[68] RADIANT INFOSYSTEMS. http://radiantinfo.com/index.html, 2007. 118

[69] OKI IRISPASS. http://www.oki.com/en/iris/. 23

[70] A. JAIN, R. BOLLE, AND S. PANKANTI. *Biometrics: Personal Identification in a Networked Society.* Kluwer, 1999. 7, 19, 20, 25, 26

[71] A. K. JAIN, A. ROSS, AND S. PANKANTI. Biometrics: A tool for information security. *IEEE Trans. Information Forensics and Security*, **1**[2]:125–143, June 2006. 13, 58, 59, 62

[72] AHMED JERRAYA AND WAYNE WOLF. *Multiprocessor Systems-On-Chips (Systems on Silicon).* Ed. Morgan Kaufmann, 2004. 77

[73] ISO/IEC JTC1/SC37. http://www.iso.org/iso/iso_technical_committee.html?commid=313770. 29

[74] GRABOWSKI K., SANKOWSKI W., AND ZUBERT M.AND NAPIERALSKA M. Focus assessment issues in iris image acquisition system. *Proc. of the 14th International Conference on Mixed Design of Integrated Circuits and Systems*, pages 21–23, June 2007. 237

[75] BRIAN W. KERNIGHAN AND DENNIS M. RITCHIE. *The C Programming Language.* Prentice Hall Software Series, 1978. 174

[76] KINGPIN. Attacks on and countermeasures for usb hardware token devices. *Proc. of the Fifth Nordic Workshop on Secure IT Systems*, 2000. 111

[77] E.P. KUKULA, S.J. ELLIOTT, R. WAUPOTITSCH, AND B. PESENTI. Effects of illumination changes on the performance of geometrix facevision 3d frs. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 331–337, October 2004. 21

[78] A. SAFIR L-FLOM. Iris recognition system. U.S. PAtent 4 661 249, February 1987. 35

[79] L1id. http://www.l1id.com/pages/37-abis-system-search-engine. 201

[80] Henry C. Lee and R.E. Gaensslen. *Advances in Fingerprint Technology.* CRC Press, 1994. 21

[81] Trong-Yen Lee, Yang-Hsin Fan, Yu min Cheng, Cia-Chun Tsai, and Rong-Shue Hsiao. Enhancement of hardware-software partition for embedded multiprocessor fpga systems. *Proc. of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, November 2007. 92

[82] Yamin Li and Wanming Chu. Parallel-array implementations of a non-restoring square root algorithm. *Proc. of International Conference on Computer Design*, pages 690–695, October 1997. 205

[83] Yanbing Li, Tim Callahan, Ervan Darnell, Randolph Harr, Uday Kurkure, and Jon Stockwood. Hardware-software co-design of embedded reconfigurable architectures. *Proc. of the 37th Annual ACM IEEE Design Automation Conference*, pages 507–512, 2000. 91

[84] Tieniu Tan Li Ma, Yunhong Wang, and Denxin Zhang. Personal identification based on iris texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**[12]:1519–1533, December 2003. 42, 46

[85] Gilles Lisimaque. Technologies for id tokens. available at: http://rack1.ul.cs.cmu.edu/tw/statessecurity/lisimaque.PDF. 9, 107

[86] Judith Liu-Jimenez, Raul Sanchez-Reillo, Almudena Lindoso, and John G. Daugman. Architecture of a search engine for massive comparison in an iris biometric system. *Proc. of 40th International Carnahan Conference on Security Technology*, 2006. 224

[87] Judith Liu-Jimenez, Raul Sanchez-Reillo, and Carmen Sanchez-Avila. Biometric co-processor for an authentication system using iris biometrics. *Proc. of 38th International Carnahan Conference on Security Technology*, pages 131– 135, 2004. 205

[88] Judith Liu-Jimenez, Raul Sanchez-Reillo, Carmen Snchez-Avila, and Luis Entrena. Iris biometrics verifiers for low cost identification tokens. *Proc. of the XIX International Conference on Desgin of Circuits and Integrated Systems*, 2004. 205

[89] Mariano Lopez, Enrique Canto, Mariano Fons, Antoni Manuel, and Joaquin del Rio. Hardware coprocesor design for fingerprint image enhancement. *Proc Of 49th IEEE International Midwest Symposium on Circuits and Systems MWCAS'06*, **1**:520–524, 2006. 87, 117, 199

[90] Michael G. Lorenz, Luis Mengibar Pozo, Judith Liu-Jimenez, and Belen Fernandez-Saavedra. User-friendly biometric camera for speeding iris recognition systems. *Proc of the 42th Annual IEEE International Carnahan Conferences Security Technology*, pages 241–246, October 2008. 40

[91] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Efficient iris recognition based characterizing key local variations. *IEEE Transaction on Image Processing*, **13**[16]:739–750, June 2004. 42, 45, 46, 47

[92] Li Ma, Yunhong Wang, and Tieniu Tan. Iris recognition using circular symmetric filters. *Proc of 16th International Conference on Pattern Recognition*, **2**:414–417, August 2002. 42, 46

[93] S. Mallat. Zero-crossing of wavelet transform. *IEEE Transaction on Information Theory*, **37**[4]:1019–1033, July 1991. 51, 168

[94] Stephane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998. 51

[95] J. Mansfield and J.L. Wayman. Best practices in testing and reporting performance of biometric devices. Technical report, NPL Report, 2002. xxiii, 16, 18, 35

[96] M. Martinez-Diaz, J. Fierrez-Aguilar, F. Alonso-Fernandez, J. Ortega-Garcia, and J.A. Siguenza. Hill-climbing and brute-force attacks on biometric systems: A case study in match-on-card fingerprint verification. *Proc. of the 40th IEEE International Carnahan Conferences Security Technology*, pages 151–159, October 2006. 62

[97] Libor Masek. *Recognition of Human Iris Patterns for Biometric Identification*. Master's thesis, School of Computer Science and Software Engineering, University of Western Australia, 2003. 42, 45, 47

[98] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D.J. LoIacono, S. Mangru, M. Tinker, T. M. Zappia, and W.Y.Zhao. Iris on the move: Acquision on images for iris recognation in less contrained enviroments. *Proceedings of IEEE*, **94**[11]:1936–1947, November 2006. 34, 38

[99] Fidelica Microsystems. http://www.fidelica.com/, 2007. 118

[100] SCM Microsystems. http://www.scmmicro.com/, 2007. 118

[101] B. Moreno, A. Sanchez, and J.F. Velez. On the use of outer ear images for personal identification in security applications. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 469–476, October 1999. 24

[102] Motorola. http://www.motorola.com/, 2007. 118

[103] Carlos Mauricio Galvis Traslavi na. Introducción a la biometría. available at: http://www.monografias.com/trabajos43/biometria/biometria.shtml. 10, 107

[104] Tammy Noegaard. *Embedded System Architecture: A Comprensive Guide for Engineers and Programmers (Embedded Technology)*. Ed. Newness, 2005. 73, 75, 77

[105] S. Noh, K. Bae, and J. Kim. A novel method to extract feature for iris recogntion system. *Proc of International Conference on Audio and Video-Based Biometric Person Authentication*, pages 862–868, 2003. 46

[106] Afzel Noorea, Nikhil Tungalaa, and Max M. Houckb. Embedding biometric identifiers in 2d barcodes for improved security. *Elsevier Computer & Security*, [8]:679–686, December 2004. 67

[107] Brian Oblivion and Kingpin. Secure hardware design, 2000. 111

[108] National Institute of Standards and Technology. Nist ice database: http://iris.nist.gov/ice/, 2005. xxiii, 31, 41, 48

[109] Lawrence O'Gorman. Comparing passwords, tokens and biometrics for user authentication. *Proc. of the IEEE*, **91**[12]:2021–2040, December 2003. 8, 108

[110] Óscar A. Chávez Bosquez, Guillermo de los Santos Torres, and José Luis Gómez Ramos. Búsqueda tabú aplicada a un problema np-completo: Generación de horarios en la dais. *Congreso nacional de informática y sistemas computacionales*, September 2005. 94

[111] Andrzej Pacut and Adam Czajka. Aliveness detection for iris biometrics. *Proc. of the 40th IEEE International Carnahan Conferences Security Technology*, pages 122–129, October 2006. 62

[112] Andrzej Pacut and Adam Czajka. Aliveness detection for iris biometrics. *Proc of the 40th Annual IEEE International Carnahan Conferences Security Technology*, pages 122–129, 2006. 241

[113] PalmSecure. http://www.fujitsu.com/us/services/biometrics/palm-vein/. 23

[114] Kang Ryoung Park and Jaihie Kim. A real-time focusing algorithm for iris recogntion camera. *IEEE TRansaction on Systems, Man and Cybernetics Part C*, **35**[3]:441–444, August 2005. 38

[115] Elena Pérez, Javier Resano, Daniel Mozons, Hortensia Mecha, and Sara Román. Función de coste dinámica para particionamiento hw/sw multiobjetivo. *Seminario Anual de Automática, Electrónica Industrial e Instrumentación*, September 2003. 91

[116] K. Piromsopa, C. Aporntewan, and P. Chongsatitvatana. A fpga implementation of fixed-point square root implementation. *International Symposium on Communications and Information Technology*, pages 587–589, November 2001. 205

[117] Luigi Pomante. Co-design of miltiprocessor embedded systems: and heuristic multi-level partitioning methodology. *Proc. of the Internation Confenrece on Chip Design Automation*, pages 421–425, 2000. 92

[118] Madhura Purnaprajna, Marek Reformat, and Witold Pedrycz. Genetic algorithms for hardware-software partitioning and optimal resource allocation. *Journal of systems architecure, Elsevier*, **53**:339–354, 2007. 92

[119] Xianchao Qiu, Zhenan Sun, and Tieniu Tan. Coarse iris classifications by learned visual dictionary. *Proc of II International Conference on Biometrics*, pages 770–779, 2007. 202

[120] D. G. Stork R. O. Duda, P. E. Hart. *Pattern Classification.* Wiley-Interscience Publication, 2000. 11

[121] Soumyadip Rakshit and Donald M. Monro. An evaluation of image sampling and compression for human iris recognition. *IEEE Transaction on Information Forensics and Security*, **2**[3]:605–612, September 2007. 56

[122] Nalini K. Ratha, Jonathan H. Connell, and Rudd M. Bolle. An analysis of minutiae matching strength. *Proc. of International Conference of Audio and Video-based Biometric Person Authentication*, pages 223–228, June 2001. 58, 62

[123] N.K. Ratha, K. Karu, S. Chen, and A.K. Jain. A real-time matching system for large fingerprint databases. *IEEE Transaction on Pattern Abakysis and Machine Intelligence*, **18**[8]:779–813, 1996. 201, 202

[124] Paul Reid. *Biometrics for Network Security.* Prentice Hall, 2003. 108, 109

[125] Raul Sanchez Reillo, Carmen Sanchez Avila, and Ana Gonzalez Marcos. Biometric identification through hand geometry measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**[10]:1168–1171, October 2000. 22

[126] J. Javier Resano, M. Elena Pérez, Daniel Mozos, Hortensia Mecha, and Julio Septién. Analyzing communication overheads during hardware/software partitioning. *Journal of Microelectronics, Elsevier*, **34**:1001–1007, 2003. 90

[127] Javier Resano, Daniel Mozos, Elena Pérez, Hortensia Mecha, and Julio Septién. A hardware/software partitioning and scheduling approach for embedded systems with low power and high performance requirements. *Lectures Notes on Computer Science*, [2799]:580–589, 2003. 90

[128] N. Ritter, R. Owens, J. Cooper, and P.P. Van Saarloos. Location of the pupil-iris boder in slit-lamp images of the cornea. *Proc. International Conference on Image Analysis and Processing*, pages 740–745, September 1999. 44

[129] A. Ross and S. Shah. Segmenting non-ideal irises using geodesic active contours. *Proc. of Biometrics Symposium*, September 2006. 44

[130] Arun Ross. An introduction to multibiometrics. *Proc. of the 15th European Signal Processing Conference (EUSIPCO)*, pages 20–24, 2007. 27

[131] Arun A. Ross, Karthik Nandakumar, and Anil K. Jain. *Handbook of Multibiometrics*. Springer, 2006. 8, 11, 13, 21, 25, 26

[132] B. Ruiz-Mezcua, D. Garcia-Plaza, C. Fernandez, P. Domingo-Garcia, and F. Fernandez. Biometrics verification in a real environment. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 243–246, October 1999. 17, 37, 40

[133] T.D. Russ, M.W. Koch, and C.Q. Little. 3d facial recognition: a quantitative analysis. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 338– 344, October 2004. 21

[134] Stuart Russel and Peter Norvig. *Artifical intelligence: A modern approach*. Prestice Hall International Editors, 1995. 93

[135] Carmen Sanchez-Avila and Raul Sanchez-Reillo. Two different approaches for iris recognition using gabor filters and multiscale zero-crossing. *Pattern Recognition*, **38**[2]:231–240, 2005. xxiii, 32, 42, 43, 45, 46, 47, 48, 49, 51, 53

[136] R. Sanchez-Reillo, B. Fernandez-Saavedra, J. Liu-Jimenez, and C. Sanchez-Avila. Vascular biometric systems and their security evaluation. *Proc. of IEEE International Carnanhan Conference on Security Technology*, pages 44–51, October 2007. 23, 37

[137] Raul Sanchez-Reillo. *Mecanismos de Autenticacion Biometrica mediante tarjeta inteligente*. PhD thesis, Polithecnic University of Madrid Department of Photonic Technology, 2000. 107, 109

[138] Raul Sanchez-Reillo. Achieving security in integrated circuit card applications: Reality or desire? *IEEE Aerospace and Electronic Systems Magazine*, **17**:4–8, June 2002. 58

[139] Raul Sanchez-Reillo, Raul Alonso-Moreno, Adam Czajka, and Kwon Y.B. Automatic remote evaluation system for biometric testing. *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008)*, 2008. 60

[140] Raul Sanchez-Reillo, Judith Liu-Jimenez, and Luis Entrena. Architectures for biometric match-on-token solutions. *ECCV Workshop BioAW*, pages 195–204, 2004. 15, 102, 108

[141] Raul Sanchez-Reillo, Judith Liu-Jimenez, Michael G. Lorenz, and Luis Mengibar Pozo. Image compact formats for iris samples for interoperability in biometric systems. *Proc of the 42th Annual IEEE International Carnahan Conferences Security Technology*, October 2008. 56

[142] Stephanie Schuckers, Larry Hornak, Tim Norman, Reza Derakhshani, and Sujan Parthasaradhi. Issues for liveness detection in biometrics. In *Biometrics Consortium Conference, 2002*. Biometrics Consortium Conference, September 2002. 62

[143] Richard S. Snell. *Clinical Anatomy of the Eye*. Blackwell Science, 1998. 32, 63

[144] Michael E. Snyder, Christopher Khang, Scott E. Burk, and Robert H. Osher. http://www.osnsupersite.com/view.aspx?rid=23421, 2007. 33

[145] Xilinx Fast Simple Link Specification. http://www.xilinx.com/products/ipcenter/fsl.htm. 213

[146] Zhenan Sun, Tieniu Tan, and Yunhong Wang. Robust encoding of local ordinal measures: A general framework of iris recognition. *Proc of International Conference on Computer Vision ECCV 2004. Available: www.nlpr.ia.ac.cn/english/irds/papers/sunzn/ECCV2004.pdf*, May 2004. 46

[147] Congent Systems. http://www.cogentsystems.com/. 15, 118

[148] Iridian Technologies. http://www.iriscan.com/. 23

[149] NIST Image Group's Iris Interoperability Exchange Test. http://iris.nist.gov/irex/, 2008. 56

[150] C. Tisse, L. Martin, L. Torres, and M. Robert. Personal identification on technique using human iris recognition. *Proc. On Vision Interface*, pages 294–299, 2002. 44, 46, 47

[151] Umut Uludag, , Umut Uludag, and Anil K. Jain. Fuzzy fingerprint vault, 2004. 68

[152] Umut Uludaq, Sharath Pankanti, Salil Prabhakar, and Anil K. Jain. Biometric cryptosystems: Issues and challenges. *Proceeding of IEEE*, **92**[6]:948–960, June 2004. 62

[153] Darren Whitley. A genetic algorithm tutorial. 95

[154] Theerayod Wiantong, Peter Y.K. Cheung, and Wayne Luk. *Design Automation for Embedded Systems*, **6**, chapter Comparing Three Heuristic search methods for functional partitioning in Hardware-Software co-Design, pages 425–449. Springer, July 2002. 93, 94, 95

[155] R. P. Wildes. Iris recognition: an emerging biometric technology. *Proceeding of IEEE*, **85**[9]:1348–1363, September 1997. 42, 45, 46

[156] Shenglin Yang, Patric Schaumont, and Ingrid Verbauwhede. Microcoded coprocessor for embedded secure biometric authentication systems. *Proc. of the IEEE/ACM/IFIP International Conference on Hardware - Software Codesign and System Synthesis*, pages 130–135, 2005. 87

[157] Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. Microcoded coprocessor for embedded secure biometric authentication systems. *Proc of IEEE International Conference on Hardware Software Codesign and International Symposium on Systems Synthesis*, pages 130–135, October 2005. 87

[158] Jinyu Zuo, Natalia A. Schmid, and Xiaohan Chen. On generation and analysis of sytetic eyes iris images. *IEEE Transaction on Information Forensics and Security*, **2**[1]:77–90, March 2007. 41, 223