A VLSI SMART SENSOR-PROCESSOR FOR FINGERPRINT COMPARISON

by

Stuart Anderson

A thesis submitted to the Faculty of Science,
University of Edinburgh, for the degree of
Doctor of Philosophy

Department of Electrical Engineering,
1993

Abstract of Thesis

Image processing techniques are increasingly being applied to new applications beyond their traditional uses for remote sensing image data enhancement. These new areas, such as machine vision for automated production line monitoring and control and financial transaction security, require low-cost compact but highly reliable systems. This thesis discusses some of the problems in achieving this goal and presents a novel approach to the implementation of low-cost real-time image processing systems.

The method presented in this thesis utilises the usual system design leverage offered by VLSI of reduced cost, power, size and weight; achieved as a result of the freedom to efficiently map algorithms to hardware. In addition, substantial further advantages are obtained by integrating the image sensor and preprocessing interface circuits onto the same silicon substrate.

During the course of this work three custom integrated circuits for real-time image processing were designed, simulated, fabricated and tested. Two of the devices form the image processing core of an entirely new, working, fingerprint based access control system. These designs then led to the development of the third device and the main focus of this thesis, a highly integrated sensor-processor for fingerprint comparison. This device has applications in many fields where personal identification is vital such as physical access control, financial transactions and health care. The architecture can also be adapted to address more general pattern recognition tasks. It is shown that through architectural enhancement of the fingerprint comparison system increased processing performance is obtained. Efficient integration of the sensing, processing and memory elements also provides the potential for reduced manufacturing costs.

Declaration

Unless otherwise stated, the material contained herein was researched and composed entirely by myself in the Department of Electrical Engineering at the University of Edinburgh, between October 1987 and March 1993.

Stuart Anderson

Acknowledgements

I wish to express my thanks to all those people who have helped in any way towards this thesis.

In particular I would like to thank my supervisor, Dr. D. Renshaw, for his technical guidance and constant encouragement throughout this work. I am also indebted to W.H. Bruce and Prof. P.B. Denyer for describing the intricacies and minutiae of the fingerprint comparison algorithm. A special word of thanks must go to G. Wang for his advice and technical support in the development of the custom image sensor.

Finally, I would like to thank Dr. Y.V. Roberts for her invaluable suggestions and comments during the final preparation of this thesis.

I conclude with an acknowledgement to the following organisations, De La Rue Plc., VLSI Vision Ltd., and the SERC, whose funding has made this research possible.

Table of Contents

Chapter 1. INTRODUCTION

1.1 MOTIVATIONS

The use of image processing techniques to solve monitoring and control problems has, until recently, been restricted to military, space and a few hazardous industrial applications where low-cost solutions were of secondary importance to system performance. The main problem has not been one of algorithmic development but the cost of the processing hardware necessary to cope with the inherently high computational requirements of image processing.

With the advances in implementation technologies over the last two decades, providing ever greater levels of processing power *per* pound, many new application areas, especially in manufacturing situations, can now be addressed. The continuing development of ever lower cost computing platforms, with increased processing capability, has reached the point where commercial and domestic image processing systems are becoming economically viable. The possible applications for low-cost image processors cover a wide range of products and functions such as traffic speed control, video-telephones, identity verification and production-line monitoring.

Hardware solutions to these types of image processing problems are normally achieved in one of two ways. The first makes use of general purpose image processors, or function specific integrated circuits, with additional glue-logic and high-level control circuitry to perform the desired function. Problems associated with this approach are numerous. They include difficulty in finding suitable devices to implement the given algorithm, or inefficient implementation due to circuit redundancy, performance bottlenecks and accuracy considerations given the limited set of available functions.

The other approach is to produce a custom solution in one or more VLSI devices using appropriate functional modules to implement the required processes. The use of custom ASICs helps to reduce the problems associated with the former approach and can produce an efficient mapping of the algorithm to an architecture optimized for area, algorithmic accuracy or speed and also has weight and cost advantages. One perceived disadvantage of the ASIC based approach is the algorithm is essentially frozen at the time of implementation and cannot be enhanced without a hardware redesign.

Typically, VLSI design methodologies have only been used to integrate the computationally intensive digital elements of image processing systems such as correlators, array operators, and data compressors. These devices only provide the data processing component of a complete image processing system.

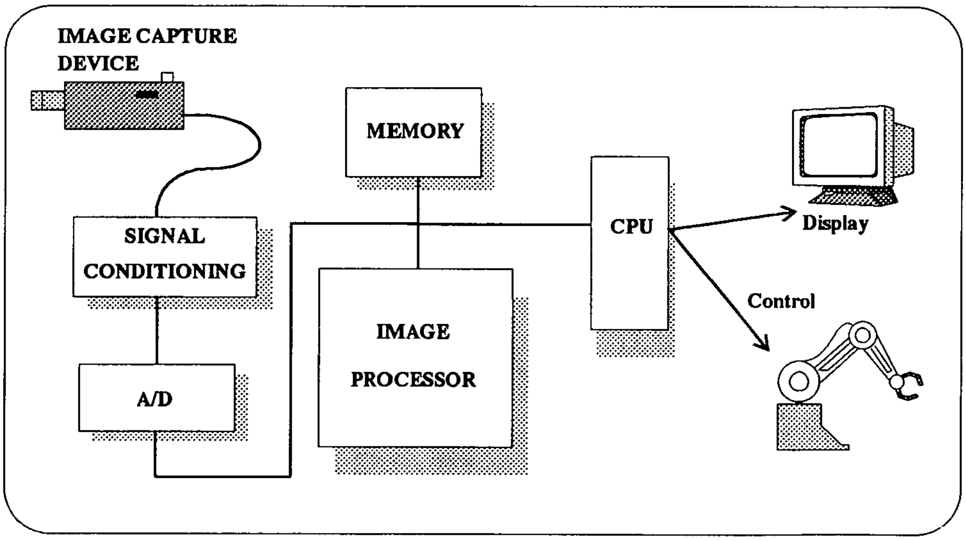


Figure 1.1 Generic Image Processing System

A generic image processing system for the analysis and recognition of images, see figure 1.1, presently consists of five main elements:- an image sensor; a data preprocessor; image analysis hardware; memory; and a microprocessor.

The input medium to this class of system is visible electromagnetic radiation which is sensed and converted into an analogue electrical representation of the image. The sensor is usually based on vidicon or solid-state CCD or MOS technology. The electrical signal often corresponds to some standard format, such as CCIR or NTSC video, in order to simplify the interfacing to various systems, or it may simply be a raster scan read-out. A preprocessor stage then performs global functions such as spatial noise filtering, analogue-to-digital conversion and image normalisation before the main image processing task is carried out. The core of the system can be either a high-performance, single-chip DSP or, for real-time operation, an array of general purpose processors or dedicated image array-processors. Memory is required to buffer images, and to store reference data and results for post-analysis. The final component of a generic image processing system is a low-bandwidth (when compared with the capability of the image processor) microprocessor which is needed to perform system control functions and analysis of results. An inherent performance bottleneck exists with systems of this type due to the serial nature of the transfer of the data from the sensor to the data processing system. The emergence a new type of VLSI device, called a smart sensor-processor, provides a possible solution to this problem and other limitations of existing image processing hardware such as cost, power consumption and size.

It is proposed that, through integration of image sensing and signal conditioning elements with the core digital processing greater architectural performance can be achieved without detriment to the usual design leverage offered by VLSI. The available architectural flexibility can be used to efficiently match the hardware to the required function. The economic and performance pay-off of this approach will provide an commercially viable method for the development of powerful, compact and reliable solutions to machine vision problems in manufacturing and enable a wide range of new consumer products based on the smart-sensor concept to be created.

1.2 OBJECTIVES

The objectives of this thesis are firstly to investigate and develop architectures for application specific image processing, including the integration of image sensors to create smart sensor-processors. Secondly, to design a hardware solution for a real-world application which will be used as a vehicle to demonstrate the technical and economic practicality of the proposed implementation technique. The design example will also highlight the degree of system integration which can be achieved with currently available VLSI technologies and integrated circuit design software. Engineering trade-off decisions covering design time, algorithmic mapping and implementation architecture will be combined to form a measure of the efficiency of a particular design approach. These elements, along with the obvious physical and performance characteristics of the integrated device, will be compared with examples of the traditional system implementation approach described earlier.

The practical work presented in this thesis will describe the design and implementation of a fingerprint comparison system for automated personal identity verification as a single chip smart sensor-processor. The choice of this application to demonstrate the viability of the proposed implementation method has been made for a number of reasons. Firstly, the system is commercially relevant. With the explosion in electronic financial transaction systems available for use by the general public, the need for secure and accurate identity verification has become vital to minimise fraud. The use of human biometrics such as handwriting, fingerprints and voice patterns are considered as potential solutions to this problem. Unfortunately, many of the algorithms which have been developed with sufficient discriminatory accuracy have had their practical use limited by hardware implementation costs.

Secondly, the algorithm forming the core of the proposed system has been well defined and a detailed description is available. Although the algorithm has been tuned for fingerprint comparison, the basic functional blocks implementing the algorithm can easily be adapted to other image processing tasks of a similar nature. An example of the algorithm's flexibility is its proposed use for fertility screening by matching ferning patterns in smear samples. This application is briefly described in chapter 8

Finally, a real-time prototype access control system based on the same algorithm has been developed which contains all the basic elements of a generic image processing system as described earlier, *i.e.* sensor, signal preprocessor, and a computationally intensive image processor. The prototype system provides a set of physical, performance and economic measurements which can be compared with those of the proposed integrated sensor-processor. This makes the fingerprint comparison application a particularly suitable demonstrator for the smart sensor-processor implementation approach.

1.3 THESIS SUMMARY

Chapter 2 briefly reviews the history of implementation methods and technologies for image processing systems highlighting the limitations of each type. Examples of current progress in the development of VLSI circuits for image processing are presented. These include general purpose image processors, function specific and application specific circuits, and smart sensors.

Chapter 3 provides an overview of the application area of the demonstration system, personal identity verification using human biometrics. A variety of different commonly used biometric features are described, comparing their usefulness for automated identity verification. A summary of commercially available identity verifiers is also provided, giving details of performance and system cost where possible.

Chapter 4 presents a brief description of the prototype fingerprint based identity verifier system and the fingerprint comparison algorithm it implements. At the core of the system are two high performance image processing ASICs. The architectures used to implement the preprocessing and comparison algorithms are given, along with details of their physical and performance characteristics.

Chapter 5 describes the architectures used to implement the functional elements of the prototype fingerprint comparison system to a single-chip smart sensor-processor. A sensor technology using standard low-cost digital ASIC processing has enabled the integrated sensor-processor design approach to be considered. The background to the development of this technology and its main design characteristics are also presented in this chapter. At the end of Chapter 5 the architectural advantages, such as increased performance and reduced memory requirements, of the single-chip design approach are discussed.

Chapter 6 briefly describes the two different VLSI design methods, full custom and silicon compilation, that have been used to implement the fingerprint comparison architecture presented in the previous chapter. The rest of the chapter details the implementation and simulation of each of the functional blocks, and their integration to form a single-chip smart sensor-processor.

Chapter 7 describes the post-fabrication test procedures used to verify circuit operation and presents the results of these initial tests on a prototype batch of integrated sensor-processors.

Finally, Chapter 8 provides a discussion of the successes and failures of the design approach as a solution to the economic implementation of real-time image processing systems using VLSI in the form of smart sensor-processors.

Chapter 2. VLSI FOR IMAGE PROCESSING

In order to set the scene for the work presented in this thesis it is necessary to review the available hardware implementation alternatives for image processing systems. This chapter provides a very brief review of hardware architectures used to implement image processing functions and surveys a range of VLSI devices designed specifically for use in image processing systems. Chapter 3 reviews biometric techniques and applications, justifying the choice of a fingerprint comparison based identity verifier as a technically and commercially relevant demonstration vehicle for the VLSI smart sensor-processor design approach.

2.1 IMAGE PROCESSING

Image processing is principally the interpretation of pictorial information for autonomous system control or the improvement of image quality. One early form of image processing involved the Trans-Atlantic transmission of coded newspaper pictures by submarine cable and their subsequent reconstruction using a special telegraph printer. This system, known as the Bartlane cable picture system, was introduced in the early 1920s. The real stimulus to the development of image processing techniques came with the space programme in the 1960s. The requirements of space exploration produced technologies enabling the construction of powerful digital computers for control systems and data analysis. Work at the Jet Propulsion Laboratory to enhance pictures returned from early missions to the moon formed the starting point of the rapid development of image processing techniques and application areas which continues today. Looking back to the Bartlane picture transmission system of the 1920s it is interesting to note that, today, one of the fastest expanding application areas for image processing is multi-media information technology *i.e.* the transmission of information and pictures. Other fields where image processing algorithms are used to enhance picture quality, or facilitate understanding of the information contained within the image include biology, archeology, geography, physics, medicine and law enforcement.

Increasingly, image processing techniques and technologies are being applied to the automatic interpretation analysis of the data within a scene, not just image enhancement for human analysis. This type of image processing technique is widely used in field of medicine where automated systems are being developed to aid the analysis of cervical smears[44], blood samples[28] and to automatically classify chromosomes[40]. Other areas where automated image analysis techniques are being utilised include military applications such as reconnaissance, weapons guidance systems, and law enforcement for automatic fingerprint comparison[11] and car registration plate recognition[32]. A further area where image processing is beginning to make an impact is in production line monitoring and quality control applications, commonly known as machine vision[34], [36], [37].

The continued expansion of image processing into higher volume commercial and domestic systems is currently limited by the cost of the processing power required to implement reliable solutions to real-world image processing problems. This situation can only be addressed by the development of appropriate architectures and technologies to allow the efficient implementation of image processing algorithms. A brief review of the computer architectures which have been used to implement image processing algorithms is presented in the next section.

2.2 ARCHITECTURES

It is has been widely stated[29],[30],[39] that no single processing architecture is capable of efficiently handling all types of image processing function. Before deciding whether or not to concur with this view it is necessary to look at some of the possible implementation methods for image processing algorithms.

2.2.1 Software

The development of software to implement image processing algorithms is the traditional method used to solve vision applications. This approach is particularly flexible, allowing modifications and enhancement of the algorithm to be made quickly and inexpensively. The major drawback of this implementation approach is the inability of general purpose, von Neumann type, computing platforms to deliver the necessary processing power to cope with the huge quantities of data generated by image processing tasks. A further penalty of such vision processing systems is that they are not capable of real-time operation, severely limiting their usefulness for practical vision applications.

One solution to the processing bottleneck of von Neumann based computer architectures is to introduce parallelism in various forms to increase data throughput. Parallel architectures are particularly useful for the implementation of low-level, pixel-oriented, image processing functions such as neighbourhood processing, edge detection and convolution functions. Classification and comparison of the vast number of different parallel computing architectures is not easy. One taxonomy which is widely used for general purpose parallel computing architectures, proposed by Flynn[27], classifies systems by the relationships between the data and instruction streams within the system. The Flynn taxonomy is also used to classify image processing system architectures. The four classifications established by Flynn are **SISD** - Single Instruction Single Data, **SIMD** - Single Instruction Multiple Data, **MISD** - Multiple Instruction Single Data and **MIMD** - Multiple Instruction Multiple Data.

2.2.2 SISD

The SISD class of machine is the conventional von Neumann type of architecture in which a single processor executes instructions on a single data set. This architecture is widely used and can be found in a large variety of computing platforms from a desktop P.C. to supercomputers like the Cray-x series. The SISD model as been extended to include pipelined processor architectures where individual processes were implemented as separate functional units. As a data set passes from one serial processor to the next, a new data set is consumed by the previous processor.

2.2.3 SIMD

The SIMD type of parallel processing architecture is particularly appropriate for implementing low-level image processing functions such as neighbourhood processing, convolution, thresholding, and edge-detection. SIMD processors are formed by constructing an array of identical processors controlled by a master. The controlling processor transmits an instruction to each processor in the array which performs the instruction on its local data. Processor to processor communication depends on the particular system but is typically nearest neighbour only. Unfortunately, SIMD systems are not particularly suitable for higher level image processing functions such as histogramming, ranking, feature extraction, or object matching.

2.2.4 MISD

Although there is little evidence of current processing systems which can be classified as MISD, architectures of this type may still be of use in image processing systems. An architecture where the incoming data stream is processed in a parallel manner by independent processors, each performing a unique function, can be envisaged. For example, this type of system could be used to extract edge, texture, shape, and image difference information simultaneously from a single input data stream.

2.2.5 MIMD

MIMD systems have an array of processors often arranged in a similar manner to those in SIMD machines, but with the ability for each processor to be executing different instructions on its local data set. MIMD systems are capable of implementing a wide range of image processing functions but with varying degrees of efficiency. Unfortunately, the effort required to write and debug programs for this class of machine and the interprocessor communication overhead limit the usefulness of current MIMD architectures according to Cypher[41]. The application independent design emphasis adopted by all these parallel processing architectures is a limitation to their economic practicality. Inherent in their generality is the inefficient mapping of specific image processing algorithms to the available processing resources. As Hummel[41] has stated in a discussion on the topic of parallel architectures for vision algorithms, "It is a shame to pay for 64,000 processors and then at any given time only use a fraction of the processors to do useful work". This observation applies to any processor whose function is linked to a specific pixel or group of pixels. This suggests that for an efficient implementation of an image processing algorithm a multi-level processing architecture is the most suitable. This type of system could include processor architectures from any of the Flynn classifications. For this reason an extension to the taxonomy is required to accommodate these multi-architectural processing systems.

2.2.6 Hybrid

Hybrid systems combine parallel array processors for low level pixel-oriented functions, special purpose devices for feature based functions (*e.g.* edge detection, object matching) and general purpose microprocessors for high level analysis and decision making. Examples of this type of hybrid multiprocessor system have been described by Sousa[42], Kioi[80] and Vellacott[45].

Although the hybrid processor architecture is seen as a solution to the efficient implementation of general purpose image processing systems, it does not provide a complete technology for the production of low-cost systems suitable for high-volume commercial or domestic image processing based products. However, when combined with cost, size, and power leverage of VLSI design techniques it could provide a method of implementing low-cost image processing solutions.

2.3 VLSI FOR IMAGE PROCESSING

The concept of multi-architectural hybrid image processing systems, combined with the design leverage afforded by the latest generation of VLSI CAD software, offers the possibility a low-cost, real-time solution to the implementation of image processing algorithms. This section reviews existing VLSI devices which have been designed for use in image processing systems. There are four main categories of VLSI image processing chips:

General Purpose - these devices implement several processing elements (PEs) and, when cascaded, form the cores of massively parallel processing systems. The PE structure can be of various types but generally consists of an ALU, datapath, registers, and possibly some distributed data memory.

Function Specific - This class of device typically implements a single general purpose image processing function such as correlation, image warping or convolution.

Application Specific - For this taxonomy, application specific devices are those which have been optimised for one particular application area such as character recognition or fingerprint comparison. The complete algorithm required to process and analyse the image data for the application is implemented as one or more custom VLSI devices.

Smart Sensor-Processors - This class of device not only includes the digital processing elements of the application specific devices but also the image sensor itself, and associated image pre-processing such as signal conditioning and digitisation.

2.3.1 General Purpose

MasPar Corporation's MP-1[31] is a typical example of a general purpose massively parallel processing architecture. The core of this system is formed from an array of processor elements (PEs) configured for SIMD operation. This architecture is typical of other, earlier, SIMD machines such as University College of London's CLIP4 (Cellular Logic Image Processor), Goodyear's MPP[38] (Massively Parallel Processor) and AMT's DAP (Distributed Array Processor) except that the MP-1 has distributed memory allowing the PEs to simultaneously access different memory locations. The MP-1 processor array contains from 1,024 to 16,384 separate units, giving the system a claimed performance of up to 26,000 million instructions *per* second (MIPS) and 1,300 floating-point operations *per* second (MFLOPS). Each of the processor elements (PEs) consists of a datapath, arithmetic logic unit, registers and in the case of MP-1, 16 kilobytes of local data memory. The array has been formed from full-custom CMOS VLSI devices, each implementing 32 PEs, and their dedicated memory.

The cost for this level of performance is in the region of £140,000 to £500,000 depending on the system configuration. Other examples of VLSI being used to implement the custom PEs for massively parallel cores of real-time image processing systems include GRID[49], SCAPE[59], DSA-1[60] and ISMP[63].

Pipeline architectures are a form of SISD processor where the successive steps of the algorithm are distributed over a number of cascaded processors. The image data is fed (usually in raster-scan format) sequentially into the pipeline to be processed.

Another architecture used for general purpose image processing is known as a dataflow architecture. Here the sequence of operations is defined by the data being processed. Examples of processor elements designed using the dataflow type of architecture have been described by Quenot[64] and Kurokawa[35].

2.3.2 Function Specific

Function specific integrated circuit architectures implement particular image processing operations. Devices have been fabricated for a number of functions including segmentation[52], motion estimation[53], image compression[70], contour line filtering[54], edge detection[50], thinning[66], convolution[56],[58],[67], template matching[25], and correlation[55],[65]. A good example of function specific VLSI is the set of eight devices from Ruetz and Brodersen[68] which implement a variety of image processing algorithms. Specifically, they implement a 3 x 3 linear convolver, a 3 x 3 sorting filter, a 7 x 7 logical convolver, a contour tracer, a look-up-table ROM, and two post processors for the linear convolver. The complete system is controlled by a SUN workstation and can work at frame rates of up to 15 *per* second on 512 x 512-pixels. More recent work by Reutz[69] has produced a set of function specific devices with the design emphasis on producing architectures with a high degree of programmability, to allow them to be used over a wide range of applications.

Beyond these function specific devices, increased levels of integration produce chips which contain several image processing functions, and are optimised for a particular class of image processing application. An example of this type of device is the OPTIC image processor[62]. Although described as a processor for use in general purpose real-time vision systems, its functionality makes it particularly appropriate for production line inspection and robotic control. The device implements filter functions (typically used for 'salt and pepper' noise suppression), dynamic thresholding and correlation with a binary template and grey-level image data. The resulting VLSI device contains 80k transistors and has been fabricated using a 1.5μm CMOS process.

2.3.3 Application Specific

Although VLSI has been used to implement particular image processing functions or groups of functions, little evidence exists of the generation of application specific image processing chips where the complete algorithm for an image processing problem is implemented as one or more custom integrated circuits. This seems to be due to the fact that most research has concentrated on developing architectures for general purpose image processing systems. As stated earlier, to enable image processing techniques to be used in consumer products, system size and cost have to fall dramatically without loss of processing performance. Only through the use of VLSI technology can all of these goals be achieved. An example of application specific VLSI devices for image processing is the two device chipset, developed by the author, implementing the core functions of the prototype fingerprint comparison identity verification system described in chapter 4 The first device performs various real-time image preprocessing and data analysis functions such as image smoothing, dynamic thresholding, and rank-value filtering while the second device implements a high throughput two-dimensional binary correlation array. Other application specific devices have been reported recently, but they integrate an image sensor as well as image processing functions and are therefore included in the following section.

2.3.4 Smart Sensor-Processors

This final classification of this brief review of VLSI image processing devices is an area in which there is rapidly increasing activity design activity. In general terms a smart sensor-processor is some form of sensor (optical, heat, pressure) integrated onto the same substrate as the data processing hardware. This review will concentrate on devices with sensors working in the visible part of the electromagnetic spectrum. An overview of smart sensor technology can be found elsewhere[82]. Integrating the image sensor with the processing functions allows the designer a great deal more flexibility in the design of efficient solutions to image processing problems. In particular it eliminates the design constraint of raster-scan data transfer between the sensor and the processor, which often results in the need for buffering of image data before consumption by the data processor. The integration of the sensor allows the designer a choice in the pixel array size, pixel aspect ratios, and sensor data read-out methods. All of these help to optimise the performance of the image processing system. Further architectural optimization opportunities exist in such systems with the removal the architectural partition between the image sensor and image processor. In sensor-processor architectures the image data pre-processing functions of the system can be more closely coupled with the image sensor.

A major barrier to the production of practical integrated sensor-processors is the incompatibility of the implementation technologies. Conventional VLSI image sensors are typically fabricated using CCD processing technology[51] while low-cost CMOS processes have been developed for the design of low-power digital processing circuits. Although it is possible to implement logic using CCD processes, the performance, complexity and cost of such circuits is not economic when compared with CMOS based logic. An example of this design approach is the CCD sensor produced by Kemeny, *et al.*[79] which includes image reformatting circuitry for some basic neighbourhood processing on the output image data. The additional circuitry occupies 2% of the active chip area. An example of device combining the qualities of CCD and CMOS technologies on the same substrate has been reported by Hakkarainen and Lee[77]. Another smart sensor-processor has been described by Kioi, *et al.*[80] which uses a novel three-dimensional processing technology, SOI (Silicon On Insulator), to create a

4-level image processing chip for character recognition. The top layer is a 5040 pixel image sensor, the next layer down implements digitisation and magnitude comparison functions, the third layer contains data buffering and mask registers, and the final layer contains template information. The device contains 0.22 million transistors on a 14.3 mm square die. Although this type of device has potential for the efficient implementation of hybrid image processors the current device densities and the high cost of the unique processing technology limit its usefulness as a low-cost image processing hardware solution.

Work in the 1980s by Lyon[81], Mead[83], and Renshaw[87],[85], among others, in developing photo-receptors and on-chip amplification circuits suitable for implementation using standard CMOS processes, has lead to the development of image sensors that are nearing the quality and performance of CCD devices. This technology has enabled the development of an increasing number of CMOS based sensor-processor devices. One of the earliest was a simple correlating optical motion detector designed by Tanner and Mead[88]. This device integrates a linear 16-pixel sensor and analogue and digital processing on the same substrate to perform correlation between successive image frames. An optical mouse was the proposed application for this chip. Another CMOS linear sensor array with processing logic fabricated on the same substrate is the LAPP[76] designed by a group at Linkoping University.

Recent work at the University of Edinburgh has resulted in highly integrated image sensors with two-dimensional photodiode arrays of up to 312 x 287 pixels incorporating on-chip sense-amplification and electronic exposure control logic by Wang, *et al.*[91]. It is particularly interesting to note that these devices are fabricated using a standard low-cost digital CMOS ASIC process. This technology is described in further detail in Section 5.2. The PASIC sensor and bit-slice processor reported by Chen, *et al.*[74] in 1990, and the MAPP2200 256 x 256 pixel sensor with on-chip A/D conversion reported by Jansson, *et al.*[78] in 1992 are further examples of the development of CMOS based sensor-processors. Other recent VLSI sensor-processors developments include the 65 x 75 pixel NCP Retina[73], a 32 x 32 pixel resistive-fuse processor[92] and a 53 x 52 pixel image contrast enhancement sensor-processor reported by Shimmi, *et al.*[89], and a neural network based cheque reader manufactured by Synaptics[43]. This final group of devices are good examples of the type architectures that can be created when implementing image processing functions as integrated mixed signal (analogue and digital) sensor-processors.

VLSI smart sensor-processors provide a flexible design technology for the implementation of hybrid processor architectures. Through the efficient mapping of image processing algorithms to silicon architectures, low-cost application specific image processing systems are now possible. The next chapter provides a review of biometric techniques and applications, justifying the choice of a fingerprint comparison based identity verifier as a technically and commercially relevant demonstration vehicle for the VLSI smart sensor-processor design approach.

Chapter 3. BIOMETRICS

3.1 INTRODUCTION

The requirement for reliable identification of individuals has grown with the rapid expansion of information technology based financial transaction services. The proliferation of automated teller machines (ATMs), electronic funds transfer at point of sale (EFTPOS) systems, home banking facilities, and telephone calling cards are examples of systems which have the need for secure identification. International travel, secure industrial and governmental facilities, sport centres, vehicle security, and logical access points (computer terminals and databases) are further areas where personal identity verification is required.

This chapter will outline a class of solution to these security problems known as *automated biometric verification,* and review its techniques, application areas and implementation technologies. The review will also provide an application context for the practical implementation of the architectural studies presented in this Thesis.

3.2 RECOGNITION AND VERIFICATION

Biometrics, in this context, can be defined as the use of any measurable human physiological or behavioural characteristic to reliably confirm a person's identity.

Through the comparison of biometric descriptors identification can be achieved by either *recognition* or *verification.* When a feature is used to determine an individual's identity from a group of known references the process is called *recognition.* Human interaction relies on this type of identification, most commonly using face and voice characteristics to determine identity and relationships. Law enforcement agencies have utilized the uniqueness of human fingerprints in solving crime for over a hundred years. Historically, this form of recognition was achieved by experts manually comparing the print recovered from the scene of the crime with those of suspects stored as a file of inked fingerprint reference cards. Today, powerful, and expensive automated fingerprint identification systems (AFIS) can compare a print with tens of thousands of references *per* second in order to determine the probable identity of the criminal.

The second way in which biometrics can be used to identify people is where it is necessary to *verify* an individual's claimed identity. Here the individual's characteristic is compared only with the reference template of the person he claims to be. The reference can be either selected from a database by some form of identifier such as a PIN, or provided by the claimant and encoded electronically on a smart card. Verification has an advantage over recognition approaches, in that the claimant has to make a conscious identity claim before verification. In certain applications (such as point of sale) this is seen as a useful additional imposter deterrent. The verification approach also has the added benefit of substantially reducing the computation required to determine identity.

The target identification application areas for automated biometric verifiers are physical access control, logical access control, personal verification. A partial list of possible applications is provided in table 3.1. All of these applications require biometric verifiers which are compact, functionally reliable, user friendly and, most of all, inexpensive.

Physical Access Control	Logical Access Control	Personal Verification
. Factories, industrial sites . Government facilities - offices - embassies - consulates . International travel . Sports facilities . Vehicle security . Military sites . Hotels . Leisure parks . Clubs	. Computers . Databases . Home banking . Pay TV	. Prisoner release . Health care - medication control - infant identification . EPOS . EFTPOS . Border control . Telephone charge cards

Table 3.1 Typical Verification Applications

3.3 AUTOMATED VERIFICATION

Before looking in more detail at specific biometric techniques and applications it is worth differentiating between *manual* and *automated* forms of identity verification. In this context manual methods typically involve the comparison of features by human visual inspection or aural comparison. In automated biometric systems feature comparison is automatically performed by electronic systems utilizing custom image processing hardware and software.

For centuries, access control and identity verification methods have relied on the allocation of a portable device or token whose ownership is verified by a secret key or password. A classic example of this token and password approach to security is handwriting, in the form of a personal signature or mark. Here, identity is verified by manually comparing the example signature on the token (*e.g.* cheque card, credit card, membership card, driving licence) with a live signature provided by the claimant. Photo identity cards and passports are another common form of manual verification device. These manual forms of personal verification are particularly prone to error or abuse, since the comparison is often performed by relatively unskilled and poorly paid personnel. Tedium is another factor which leads to higher error rates.

With the increasing use of ATMs and other unmanned self-service transaction systems, the requirement for reliable remote verification has become much more important. In this type of application magnetic strip cards combined with personal identification numbers (PINs) are the most widely accepted method of access control.

The existing approaches to identification all have their weak points; cards can be stolen; signatures easily forged; and passwords and PINs forgotten. For these reasons present methods are deemed inadequate. Automated biometric verification technologies are an attempt to create a reliable solution which can easily be tailored to fit a wide variety of security applications.

Biometric systems can additionally provide an audit trail through data logging, of both genuine and imposter access attempts. In the case of fingerprint systems, the captured fingerprint can be used in subsequent criminal investigations (although this feature is not normally advertised to users, legally they must be told).

3.4 BIOMETRIC VERIFIERS

Having outlined the need for low cost, automated biometric verifiers some of the typical features of such systems can be described. Automated biometric identification systems, in general, consist of a mechanism which allows for the sensing and capture of a particular feature, and a processor which compares the 'live' image with a previously stored reference in order to verify a user's identity (see figure 3.1). The reference is selected from a database by a PIN, or can be provided encoded on a smart card. The comparison is performed by the processor and an authentication decision is made. The claimant is then granted or denied access, as appropriate.

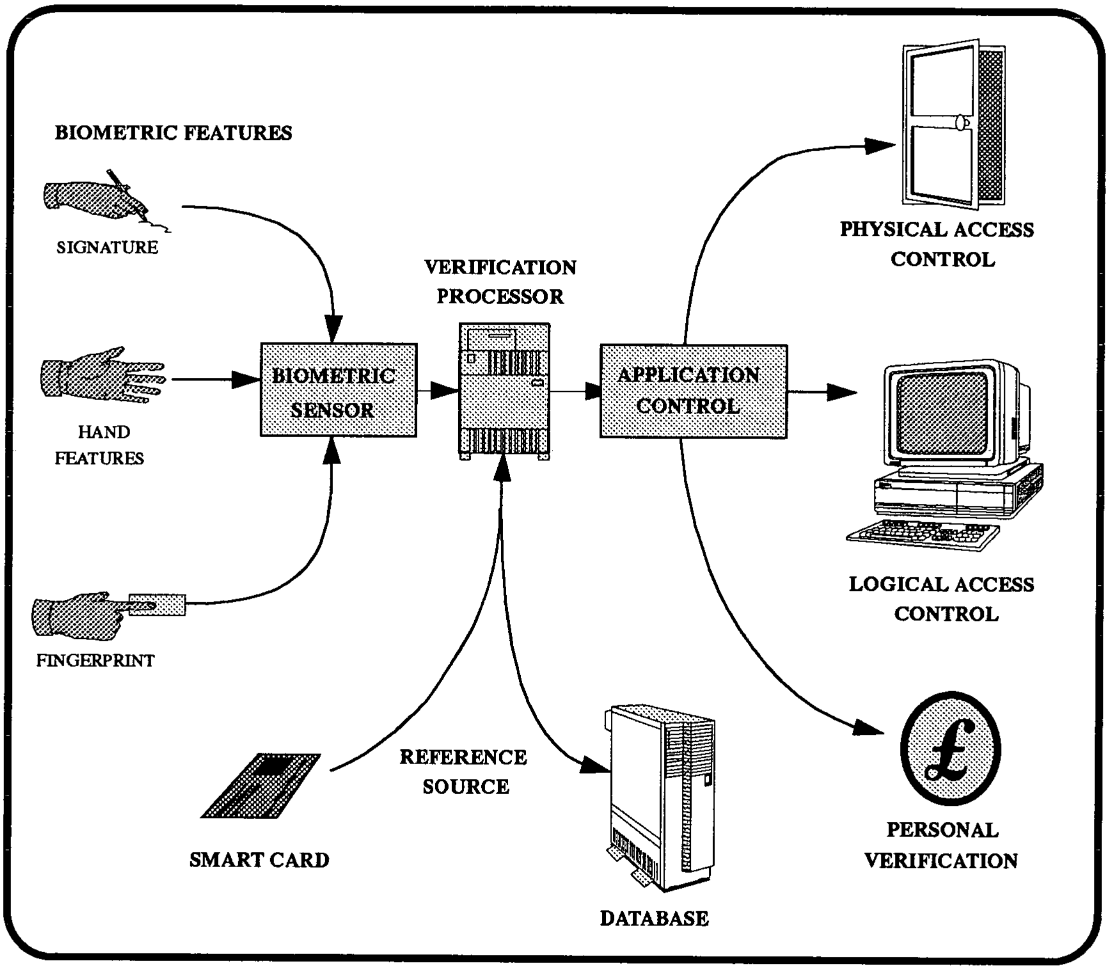


Figure 3.1 Generalized Biometric Verifier

Enrolment is the procedure performed to register or validate a new user on a system. It involves the repeated capture (typically 2 - 5 times) of a particular biometric characteristic, in the presence of a trained enroller. The captured feature is processed to create a reference template for the user, which is either stored in a database or encoded on a smart card.

A powerful feature used in some biometric systems is the ability to automatically update the reference template for a user, without having to repeat the enrolment process. This adaptive update of references is particularly useful for accommodating minor biometric variations in voice and signature based systems. The update procedure works by tracking the changes between the reference template and the presented live feature. Assuming access is granted, the template can then be modified to more closely match the user's characteristic. The configuration of the main elements of a biometric verifier; the sensor, comparator, controller, and database, is dependent on the needs of the application and available implementation technologies.

Historically, according to Driscoll[2], biometric systems were designed round a centralized architecture, due to the magnitude of the processing task in comparing a biometric sample and reference. In this configuration, the reference template database, comparison processor, and application controller are all part of a central computing resource with only the biometric sensors located at the various access or verification points. With the development of more sophisticated image processing algorithms and new implementation technologies like VLSI, a distributed architecture can now be created. Here, each access point has a dedicated sensor, database, comparator and controller which can operate in isolation, or can be networked to allow data logging or global data updates.

Another feature incorporated into some systems is the ability to determine whether or not the presented biometric characteristic is live. For fingerprints, a simple skin temperature and non-intrusive blood oxygen test is used in one system[4] to reject severed fingers or latex prostheses.

3.5 PERFORMANCE MEASURES

The performance and comparison of biometric devices is normally based on two measures, the false rejection rate (FRR) and the false accept rate (FAR).

The FRR (also described as a Type 1 error, or insult factor) is the ratio of valid attempts (or bids) to the total number of bids. While the FAR (or Type 2 error) is the ratio of successful imposter attempts to the total number of imposter attempts. Most systems allow for the adjustment of the balance between FRR and FAR. If the security against illegal acceptance is increased the likelihood of a genuine user being rejected is also increased. The converse is also true.

Another measure of comparative performance often quoted is the equal error rate (EER), which is the point in a biometric verifier's error curves where FRR = FAR. Unfortunately, the measurement of these ratios and their relationships are not very well defined, which makes for unreliable system comparison based on manufacturers' claimed performance. Directly comparable trials on identical bench marks are very costly. Trials by the Sandia National Laboratories[5],[10],[9], a U.S. government sponsored agency, provide most of the reliable published performance information on existing, commercially available, biometric systems.

The FRR for a particular system can be improved through a combination of careful enrolment, user training, and the use of adaptive template updating. The elimination of errors due to the mistyping of PINs can be achieved by using magnetic or smart cards.

Table 3.2 Commercial Biometric Verifiers

COMPANY	PRODUCT	BIOMETRIC	TYPE	PERFORMANCE			VERIFY	TEMPLATE	UNIT	COMMENTS
				EER%	FRR%	FAR%	secs.			
Fingermatrix	Mint 21 Mint11	Fingerprint	P L	- -	0.5 0.5	0.001 0.001	<3 <3	400 bytes/finger 400 bytes/finger	\$3,500 \$3,500	Error rate source
Identix	Touchlock Touchsafe	Fingerprint	P L	- -	1.8^{C} 1.8^{C}	0^{C} 0^{C}	1-2 1-2	1 kbytes/finger 1 kbytes/finger	\$3,500 \$1,895	Error rate source
Thumbscan	301	Fingerprint	L		0.5	0.01	6-10	-	\$1,195	Error rate source
TMS	Eagle, Hawk, Falcon	Fingerprint	V	-	-	-	3	24 bytes	\$2,000	No live capture Inked cards
Alpha Microsystems	TIS RACS	Voice	L,C P,C	6.5^{A} 6.5^{A}	5.1^{C} 5.1^{C}	2.8^{C} 2.8^{C}	- -	~ 8 kbyte/user ~ 8 kbyte/user	\$20,000 \$20,000	Error rate source
VoiceTek	Alpha V Alpha VI	Voice	P,C P,C	- -	- -	- -	- -	- -	\$19,500 \$70,000	8 doors >500 doors
International Electronics	VoiceKey	Voice	P,V	8.2^{A}	4.3^{C}	0.9^{A}	2	160 bytes/word	\$1,190	Error rate source
Eydentify	Model 8.5	Eye - Retina	P,L	1.5^{A}	0.4^{C}	0^{C}	1.5	40 bytes	\$4,995	Error rate source
Pideac	MarkIV	Hand Geometry	P	-	-	-	-	2 kbytes	\$3,500	
Recognition Systems	ID-3D	Hand Geometry	P	0.2^{A}	$<0.1^{C}$	0.1^{C}	2	9 bytes	\$2,000	Error rate source
Hand Scan Technologies	PG2000	Hand Print	P	-	-	-	-	10,000 points	\$9,750	
Digital Signatures	Sign/On	Dyn. Signature	P,L,V	-	2.1^{C}	0.7^{C}	3	84 bytes/sig.	From \$640	Error rate source
Cheque Alert	DigiScan	Static Signature	V	-	10	1-15	-	18 bytes	\$995	Error rate source

Key: P - Physical access control, L - Logical access control, V - Personal Verification, C - Centralized verification processor and database
EER - Equal Error Rate, FRR - False Reject Rate, FAR - False Accept Rate, A - 1 try, B - 2 try, C - 3 try

The FAR is influenced by the particular biometric descriptor being used, and the ability of the verifier to extract and compare its inherent discriminating characteristics. Examples of the performance measures (EER, FAR, FRR, verification times) for several commercial verifiers are given in table 3.2. The most secure systems are based on fingerprint and retina scan. Both return FARs approaching zero under trial conditions. The FRR for the same systems is in the region of 0.5% to 2%. In contrast the worst scores are for a signature geometry based verification system from *Cheque Alert Inc.*[11] with a FRR = 10% and FAR = 1%-15%. The human voice is a difficult biometric to discriminate accurately and consistently, and this is reflected in typical system FARs of 1%-2% and FRRs of around 5%.

3.6 BIOMETRIC DESCRIPTORS

A wide range of biometric features are utilized in such systems which can be classed as either physiological, or behavioural characteristics (see figure 3.2).

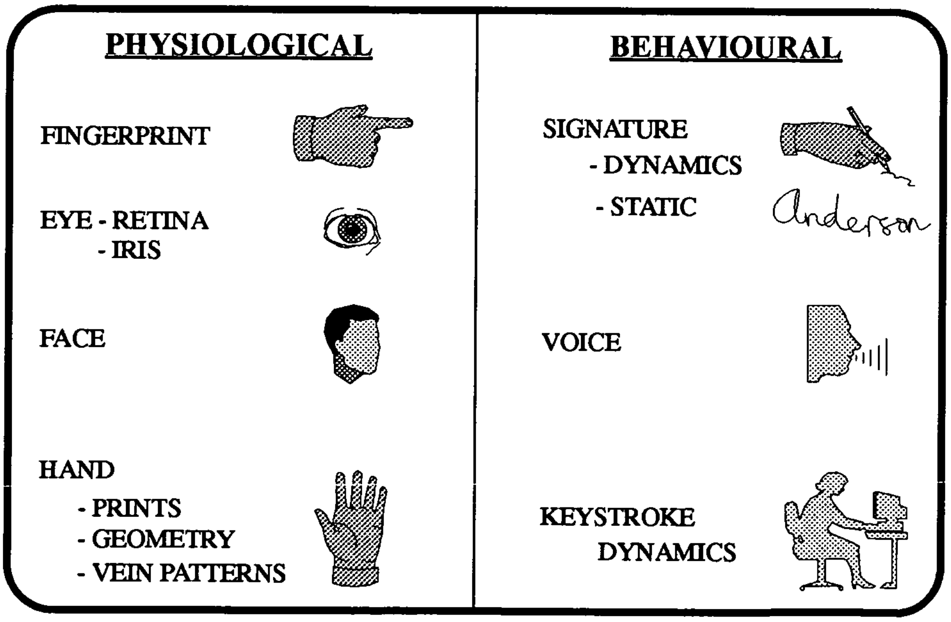


Figure 3.2 Common Biometric Descriptors

Commonly used physiological descriptors include fingerprints, hand geometry, faces and eyes. These features are basically invariant over time, except through accidental damage.

Behavioural characteristics, such as speech and handwriting, are a combination of physiological and psychological influences. The characteristics of a voice are not only dependent on a person's size, sex and heredity, but also on their accent, emotional state, and even on their state of health *e.g.* whether or not they have a cold. The remainder of this section will briefly survey the biometric descriptors which are presently used in commercial standard verifiers. Most of the information given below regarding commercial biometric verification systems has been reported by Miller in Personal Identification News (PIN)[11],[12],[13] and in papers by Parks[14],[15].

3.6.1 Eye - Retina

The blood vessel pattern on the retina and the iris pattern of the human eye are both used as biometric descriptors. *Eydentify Inc.*[11] manufactures products based on the retina pattern for both logical and physical access control applications. To gain access, the user positions his head over the unit, focuses on an alignment target consisting of a series of concentric circles, then presses a button to trigger the scan sequence. A low-intensity infrared light source is directed in a circular scan centred on the back of the retina. The reflected light is then sampled, and processed to produce a unique 40 byte template. The sensed pattern relies on the differences in reflectivity of the retinal blood vessels and surrounding tissue. The matching process is performed by an algorithm based on Fourier cross-correlation with phase matching (to compensate for rotation of the user's head) running on a Motorola 32-bit microprocessor.

3.6.2 Eye - Iris

A variation on the retina scan system is being developed by *Eye-D*[11] which uses the iris pattern of the eye, as opposed to the retina. A feed-back system of lights is used to adjust the pupil size before scanning. This type of system is only likely to be used in high security sites, since the quality of the optics and engineering required precludes low production costs.

3.6.3 Face

The use of the face as a biometric identifier is an obvious one since this is how humans perform recognition of individuals. Neural networks are viewed as the most effective method of comparing faces. Research work on a pattern recognition techniques at the Paul Scherrer Institute in Zurich (PSIZ)[1], using edge direction vectors, has been applied to face recognition with some success. The problem of impersonation by the use of photographs is one which will have to be addressed before face recognition will be acceptable for unmanned access applications. There has been much research activity[3],[18],[20] into face recognition but, as yet, there has been no commercial exploitation. A system designed to detect potential bank robbers as they enter a bank branch, developed by *NeuroMetric Systems*[12], is likely to be the first commercial use of face recognition.

3.6.4 Fingerprint

The uniqueness and stability of fingerprints is the longest established biometric, making them particularly appropriate for identity verification. Traditional non-automated methods of fingerprint comparison have been based on the classification of minutiae. Minutiae are the small characteristic whorls (spirals), bifurcations (forks) and loops, which form a unique fingerprint pattern. This technique has been adapted for use in fingerprint-based, automated identity verifiers from *Identix Inc.* and *Fingermatrix Inc.* Other methods which have been applied to fingerprint verification include the use of Fourier analysis (*France Telecom*[11] in Caen), spatial correlation (*University of Edinburgh*[24]) and neural pattern matching (*Net-ID Inc.*[13] and U*niversity of Essex*[6]). Novel approaches to the capture of fingerprint images through the use of tactile sensors (*Tactile Technologies*[11]) and high-frequency ultrasound (*Niagra Technology Labs*·[13]) are also being investigated.

3.6.5 Hand

The first commercially available, automated biometric device was installed at *Shearson Hamil* on Wall Street in 1973. The device, called Identimat, used the lengths of four fingers as a means of discrimination. Although no longer available in its original form, the principle has been developed into a three-dimensional system marketed by *Recognition Systems Inc.* Another product, from *Pideac Inc.*, also utilizes hand measurement as the basis of identification. This product differs from Recognition Systems' by using software, rather than guide posts or stops, to compensate for variation in hand placement and finger spread.

Creases on finger joints and palms are also used as physiological biometrics. A system from *Biometrics Inc.* uses infrared sensors to detect the pattern of creases on the underside of finger joints to form a 'human bar code'. The creases and lines in a area 1.5" x 1.5" on the palm of the hand is used by a biometric identifier marketed by *Hand Scan Technologies.*

An algorithm, called Veincheck which is based on the blood vessel pattern on the back of the hand has been developed by *Cambridge Consultants Ltd.*[7]. The hand is illuminated by a tungsten light source, and an image is captured using a CCD camera fitted with an infrared filter. A hexagonal grid is superimposed on the vein pattern and a connectivity matrix is produced. The comparison is performed by analysing matrices for like connections. Hand-feature based approaches are claimed to be particularly appropriate for high throughput access control applications.

3.6.6 Keystroke Dynamics

This behavioural descriptor is based on the unique rhythm with which typists enter certain groups of characters. It is claimed that as few as a dozen keystrokes are sufficient to identify a user. The principle is not new; in the days of telegraph using Morse code, operators could identify each other from their typing idiosyncrasies.

The principle of keystroke dynamics has been specifically developed for signing on and off procedures in logical access control applications. It is also the only biometric (with the possible exception of face recognition) which has the potential for continuous assessment of the operators identity. This is achieved by monitoring for frequently entered sequences of characters, and prompting the typist to re-enter a password if the system suspects a change of user. Although appropriate for database protection, keystroke dynamics does not provide a suitable solution for physical access control or ATM security.

3.6.7 Signature Dynamics

In business and financial transactions, personal signatures are by far the most widely used method of identity verification, so it is natural that they should be exploited in automated verification systems. Static signature verification is where only the graphical information contained in the signature is used for comparison. It is interesting to note that the error rate performance is the worst of all biometric systems and is only commonly found as part of bank automated credit and cheque clearing operations. These systems are similar to AFIS systems where the emphasis is on high throughput (tens of thousands of comparisons *per* second) rather than on accuracy and low cost. One company, *Cheque Alert Inc.* produce a signature geometry based system, but with a FAR of between 1% - 15%, depending on the skill of the forger, it is unlikely to provide a successful solution for point of sale applications.

Signature dynamics provide a more robust form of biometric identifier as they are much more difficult to forge and, hence, many companies have developed systems based on this biometric. As well as graphical information comparison, signature dynamics use behavioural spatial and timing information captured while the signature is being written. These systems typically require some form of sensitive writing surface and/or a special hardwired pen to detect the accelerations, directions and forces generated while the signature is being written[21]. The use of custom pens render them susceptible to malicious damage, making them suitable only for monitored applications such as point of sale or bank teller transactions.

3.6.8 Voice

Voice based verification is particularly attractive because of its acceptability to a wide range of users. Peckam[16] classifies voice analysis systems as either text-dependent or text-independent. Text-dependent systems require a previously specified word to be spoken which is then matched with stored samples of the same word throug non-linear time alignment. In text-independent verification, acoustic measurements are taken from a characterization template for an individual speaker. These descriptors include pitch, formant frequencies, linear prediction coefficients, energy measurements and Fourier Transforms.

Voice verification systems are usually configured in one of two ways. They can employ custom voice sensor hardware (a telephone!) and a speech analyser at each point of access, stand-alone or networked. Alternatively they may use a centralized voice processor and with a reference database accessed *via* the existing telephone networks. Tape fraud is a recognized problem with voice verifiers. A partial solution for text-dependent systems is to use random word selection from a fixed library, but this has a penalty of increased enrolment times which may not be acceptable. Voice-based systems do have the potential for hands-off operation which could be useful for access control applications. Companies currently marketing voice based products include *Electronic Warfare Associates, Voice Sciences, Alpha Microsystems, International Electronics (*formerly *Ecco), VoiceTek (*formerly *Voice Control Systems*).

3.7 FUTURE DIRECTIONS

The cost of biometric verifiers in the eighties have fallen from around £7,000 *per* unit in 1985 to around £1,500 in 1989. According to data published in PIN[11], this rapid decline in system prices has stabilized over the last two years with the average price dropping by only £300 during 1990. At this level biometric verifiers are only suitable for higher value industrial, military, and government applications. To expand the use of biometric products into new markets, such as personal verification for EFTPOS, and domestic applications like vehicle and home security, the unit price will need to fall to the region of £100-£200. This price reduction will be met by a number of improvements to current technology.

One change, applicable to systems targeted at EFTPOS applications, will be the elimination of the requirement for either a local or centralized database, and consequently the need for a PIN, through the use of smart card technologies. The simplest form of smart card or IC card contains some nonvolatile memory which can be accessed by an external reader. Rather than storing a reference in a local database for each valid user, it is encoded on a smart card which is issued to the user. To gain access the user no longer needs a PIN but simply inserts the smart card into the verifier and, when prompted, provides his biometric feature for verification against the smart card reference. This is particularly useful for remote applications like credit card verification, and unmanned situations like ATMs. More sophisticated cards[17],[8] incorporate microprocessors and memory which can perform, on a single card, the functions of several credit cards, debit cards, or cheque guarantee cards, and can also act as an electronic purse.

The error rate performance requirements for the target applications are already being met by most of the existing commercial systems with average FRRs near 1% and near zero FARs for fingerprint, retina scan and hand geometry based systems. The average verification time is currently around 2 secs. In future systems a comparison time of around 0.5 secs. will be perceived by the user as almost instantaneous.

The required system cost reduction and increased performance for the next generation of biometric verifiers will be achieved through the use of application specific VLSI technology. This chapter has described some of the wide range of applications for automated identity verification systems. It has also shown that fingerprints are a particularly useful biometric feature for identity verification, providing a high level of discrimination and, therefore, security. For these reasons the choice of an automated fingerprint comparator provides a practical, commercially relevant demonstrator for the image processing system implementation approach presented in this Thesis. The next chapter describes the fingerprint comparison algorithm and a prototype identity verifier system based on it.

Chapter 4. FINGERPRINT VERIFICATION

4.1 INTRODUCTION

In order to set the scene and provide a framework for the image processing case study presented in the remainder of the thesis, this chapter will outline the algorithm which has been developed at Edinburgh University for capturing and comparing fingerprint patterns. A full and detailed description of the algorithm and its development is available elsewhere[24],[22],[26]. A prototype fingerprint verification unit (FVU) for physical access control has been developed[46]. This system was chosen because it represented a practical, complex, and industrially relevant example, for which all the details of function and structure were available and well understood.

At the heart of the unit is an ASIC-based subsystem which implements the Edinburgh fingerprint comparison algorithm. This ASIC chipset[47],[48] was the precursor to the highly integrated image sensor-processor presented later in this thesis. A brief description of the ASIC architectures and the chip performances will be presented in section 4.4.

4.2 FINGERPRINT VERIFICATION UNIT

The FVU[46] is divided into several functional areas as illustrated in figure 4.1. The video processing system consists of a CCD camera module, a CCD to ASIC interface board, black level clamp circuitry and a 7-bit A/D convertor.

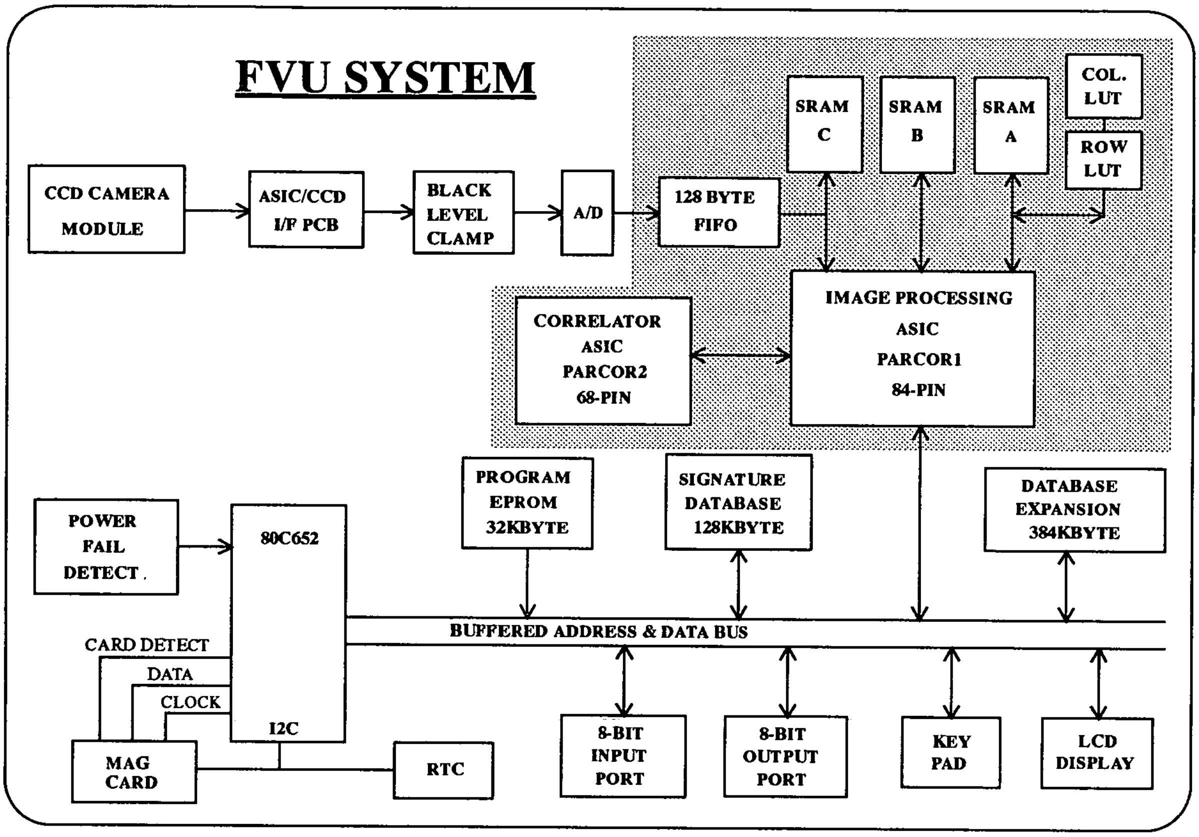


Figure 4.1 FVU Functional Schematic

The interface board generates signals which are used to sample a particular window within the complete image produced by the camera module. The black level clamp circuitry is used to scale the signal for input to the A/D convertor. The output from the A/D is then buffered using a 128 byte FIFO. This is required to even out the 'bursts' of pixel data generated by the camera module.

The delay between image lines is used by the first of the two custom ASICs, Parcor1, to empty the FIFO before the next line is digitized. Parcor1 preprocesses the grey-level data, performing filtering and thresholding functions, to produce a normalized black and white representation of the fingerprint image. The binary image is then compared with a pre-stored reference image using an algorithm based upon a modified correlation process. The comparison process is controlled by Parcor1 which sequences the flow of image and reference data to the second ASIC, Parcor2, which performs the correlation function.

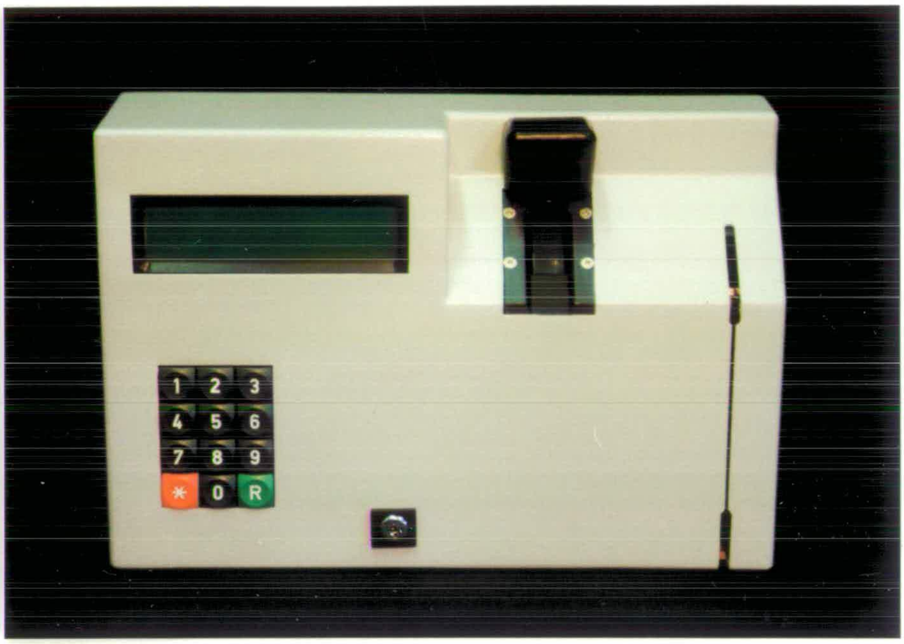


Figure 4.2 FVU General External View

Based on the results of this comparison, a decision is made as to whether or not the captured image sufficiently matches the reference template to allow access. Parcor1 also has an interface which allows communication with an 8-bit microcontroller. The microcontroller provides high-level system control functions linking the ASIC image processing subsystem to the peripheral human interface circuitry (*e.g.* LCD display, keypad, magnetic card reader). The main processor board also contains 128kbytes of RAM for storing reference fingerprint data (signatures) and other system variables. The system control software, for the embedded microcontroller, is stored in a 32kbyte EPROM.

The user interface consists of keypad which allows a user to enter a PIN code, a magnetic card reader giving an alternative means by which a user can be identified and a LCD display for the presentation of system status messages and user instructions. There is also a finger guide to help position a finger on the fingerprint sensing platen. Figure 4.2 shows the general external make-up of the prototype FVU. Other peripheral circuitry includes a serial communications port (for networking several FVU systems), a real time clock, power fail detection and 8-bit parallel input and output ports.

Table 4.1 provides a summary of some of the physical and performance characteristics of the prototype FVU. This data provides a benchmark to compare the performance of the FVU with the integrated sensor-processor ASIC described in chapters 5 and 6.

FINGERPRINT VERIFICATION UNIT	
No. of ICs (VLSI, LSI)	16
No. of ICs (MSI)	37
No. of discretes	400 (approx.)
Power	6W
Weight	2kg
Clock frequency	12MHz
Size (w x h x d)	26 x 18 x 12cm
Template size	512bytes/finger
User database	96
Match time	1.5s (typical)

Table 4.1 FVU Physical and Performance Characteristics

4.3 EDINBURGH ALGORITHM

The Edinburgh skin pattern recognition method is intended for applications where personal verification is required, such as in physical access control, credit card transactions, computer security *etc*. It is not intended for forensic work, where a sample print is used to search a database for identification of an individual. In our system, a person is identified by comparing his captured fingerprint with a previously stored reference print. This reference fingerprint signature can be accessed by the use of a PIN number from an on-line database or downloaded from a personal smart card (figure 4.3).

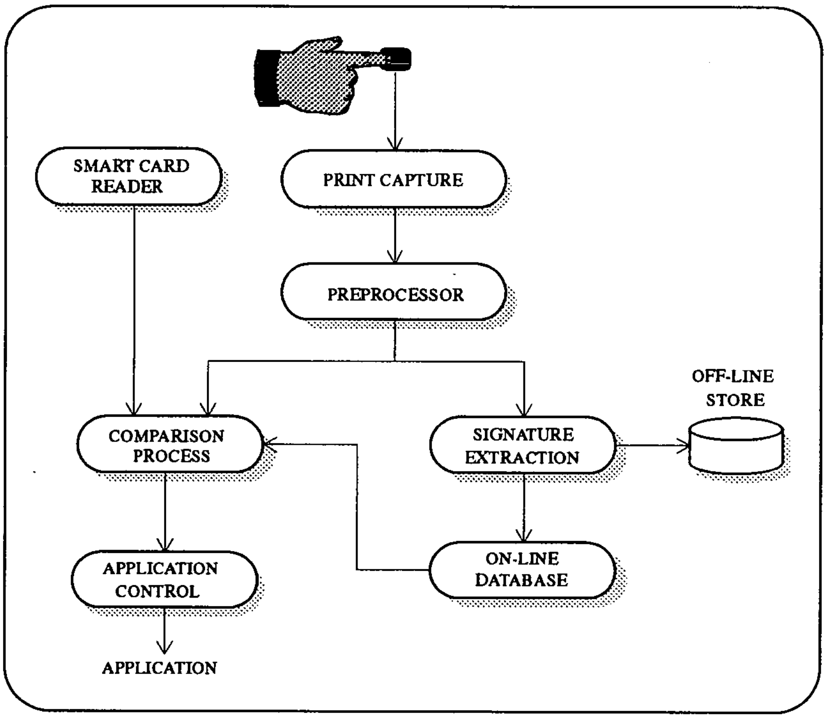


Figure 4.3 Comparison Process Organization

The main processing stages which form the skin pattern matching method are:

- **Print Capture:** The optoelectronic front-end which generates a digitized grey-scale representation of a fingerprint formed by pressing a finger on a glass surface.

- **Preprocessing:** Takes the grey-scale image and produces a normalized binary representation of the peaks and troughs.

- **Signature Extraction:** This provides the reference fingerprint image against which subsequent prints are compared.

- **Pattern Matching:** The matching technique is based upon a form of direct correlation which has been modified to allow for distortion of the fingerprint image caused by rotation, translation, and stretching.

4.3.1 Print Capture

The print-capture subsystem is required to produce a grey scale (typically 8-bit) image of 256 x 256 pixels corresponding to an area of approximately one square centimetre of the skin surface. How this is achieved, in practice, is irrelevant to the subsequent matching process provided that the same specification is used. The optical system described here has been developed and used successfully with the prototype FVU system and is also suitable for use with the single-chip version.

The simple optical system shown in figure 4.4 produces a high-contrast image of a fingerprint by utilizing the total internal reflection effect of a prism[101]. The imaging surface is the hypotenuse face of a glass prism with one of the perpendicular faces blacked out. In this configuration total internal reflection ensures that no light is projected through the remaining face of the prism. When an object, in this case a fingerprint, is pressed against the image surface the total internal reflection pattern is perturbed and incident light at the points of contact is scattered. This produces an image based on the fingerprint ridge pattern which can be viewed through the unblackened perpendicular face of the prism.

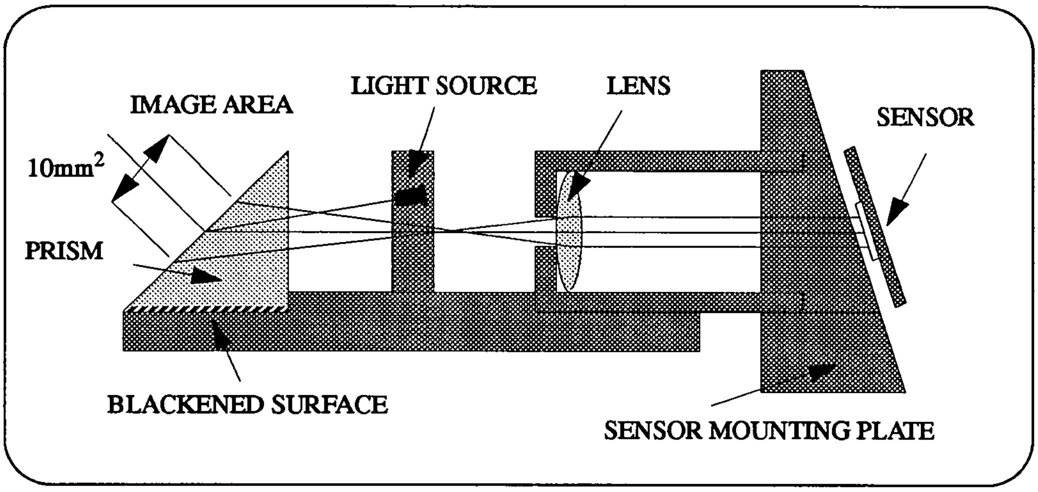


Figure 4.4 Typical Optical Setup

The image is focused onto a two-dimensional solid-state sensor, typically a CCD or CMOS device. This analogue representation of the image data is sampled, and digitized to give the required 256 x 256 pixel grey-level data ready for the preprocessing functions, filtering, and thresholding.

4.3.2 Preprocessing

The first stage of preprocessing is the application of a simple smoothing function. This is designed to remove 'speckle' noise from the image *i.e.* interference of a higher spacial frequencies than those of the ridge/trough patterns of a fingerprint. The process is achieved by convolving the image data with a 3 x 3 unary weighted window to produce an average of each pixel with its eight nearest neighbours. Non-unary weights were considered but the unary weighted template offered acceptable performance and simplified the hardware implementation.

The second part of the preprocessing function delineates the ridge/trough patterns, normalizes the image to compensate for uneven lighting and image intensity, and compresses the data to 1-bit *per* pixel. The adaptive thresholding technique requires that the 256 x 256 image is divided into 256 separate 16 x 16 sub-images or ‘patches’. For each patch, a threshold is calculated and then applied to the appropriate patch data to assign each pixel as being in a trough (black) or a ridge (white). Typically, the level of the threshold is set so as to divide the image into equal numbers of black and white pixels.

4.3.3 Signature Extraction

The matching process requires a reference signature to be compared with the captured fingerprint image. This signature is derived during the initial enrolment session and characterizes the skin pattern to emphasizes the main ridge/trough patterns of the fingerprint.

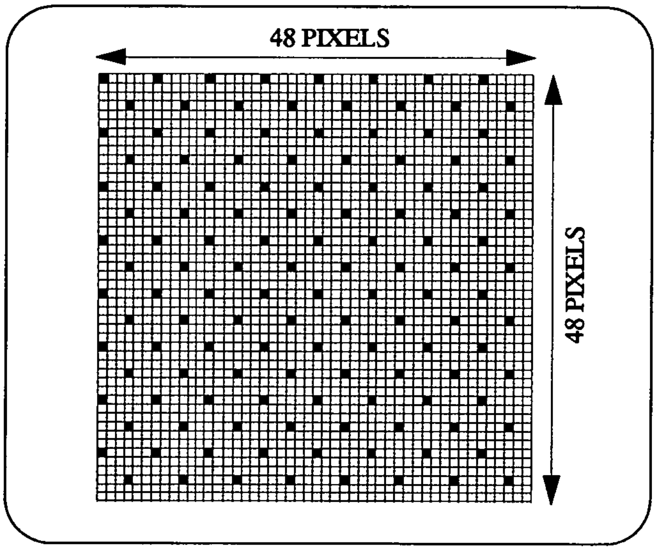


Figure 4.5 Typical Signature Template

This is achieved by applying the threshold function twice with two different threshold levels. The first threshold divides the pixels into two sets, 25% black and 75% white. A second application of the process with a different threshold divides the image into sets consisting of 75% black and 25% white pixels. The results are then analysed to assign one of three values to each pixel: *white*, *black*, or *don't care*. *White* pixels are those that were classified as white after the application of both thresholds. Similarly, pixels which were black using both thresholds are assigned as being *black*. All other pixels are labelled as *don't care*.

The signature is then extracted from the resulting 256 x 256 ternary valued image. A sparse template is positioned over a predetermined region and the sampled points are recorded to produce the fingerprint signature. The template size, number of sample points, and number of signatures can be varied for each implementation. For this application, the frame size of 48 x 48 pixels has been chosen containing 128 sample points arranged as shown in figure 4.5. This sub-sampling reduces the reference template storage requirements and decreases the computational load during the comparison process.

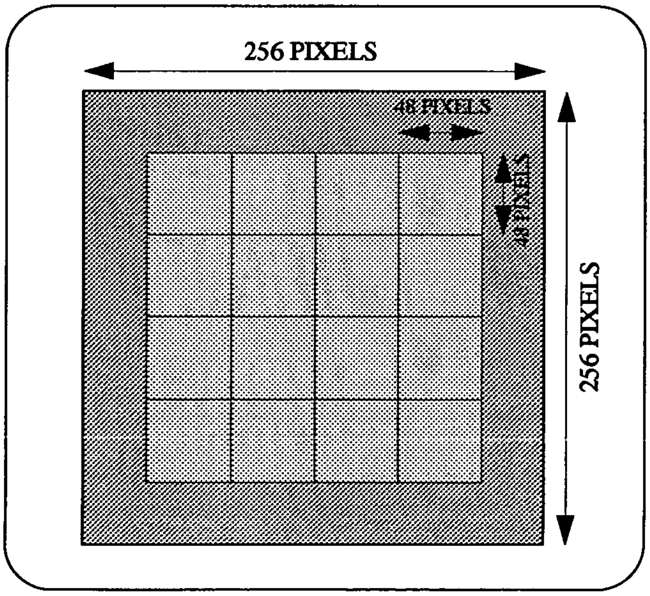


Figure 4.6 Sub-signature Template Positions

The frame is applied to the image in 16 positions to derive a set of 'sub-signatures' for each fingerprint. Figure 4.6 shows how the 16 sub-signatures are arranged in a 4 x 4 grid to cover the central region of the image. Thus the total storage for the signature for one fingerprint is 128 x 2 x 16 = 512 bytes.

4.3.4 Pattern Matching

The matching problem is to determine whether or not the captured fingerprint corresponds to the set of reference 16 sub-signatures. Each sub-signature is compared with the captured image and a correlation or similarity score is derived. This correlation is performed repeatedly for a number of positions and angles of rotation to accommodate movement between the captured and reference prints.

The highest scores for each sub-signature are ranked along with their normalized position address in a 32 x 32 grid of 8 x 8 pixel blocks covering the image space. These results are then analysed in the final part of the matching process, called polling, to make the match/fail decision.

The criterion for acceptance is that the scores for a predetermined number of sub-signatures should be present in the ranking table for any 2 x 2 area of the 32 x 32 block grid. The polling threshold used to determine a good match is based upon the maximum number of sub-signatures that can legitimately score at that position in the grid. For a centralized position all 16 sub-signatures are valid. Sufficient lateral or vertical movement reduces the number of sub-signatures from 16 to either 12 or 9. For each maximum, a polling threshold is set to discriminate between matching and rejection. The thresholds currently used in the Edinburgh algorithm are tabulated in table 4.2. A comparison decision is made based on the poll score, and a higher level control function is executed based on the result.

No. of valid sub-signatures	16	12	9
Polling threshold	10	8	7

Table 4.2 Polling Thresholds

It should be noted that the algorithm implemented in VLSI form differs slightly from the general description summarized in this section. The algorithm has been presented in a serial form which would not be practical for implementation as a real-time system. The use of parallel architectures are required to reduce the massive processing overhead of this algorithm. The variations will be highlighted as the architectures used to implement the algorithm in VLSI form are described in the next section.

4.4 ASIC FUNCTIONS AND ARCHITECTURES

A real-time image processing ASIC chipset has been designed and fabricated implementing the fingerprint matching algorithm described in the previous section.

4.4.1 Overview

Figure 4.7 shows the main elements of the fingerprint matching subsystem. Parcor1 is a real-time image processing ASIC which performs all the image preprocessing, data sequencing and analysis functions which form the fingerprint matching algorithm. The second ASIC, Parcor2, implements a high throughput, parallel binary correlation array which is used to accelerate the comparison process.

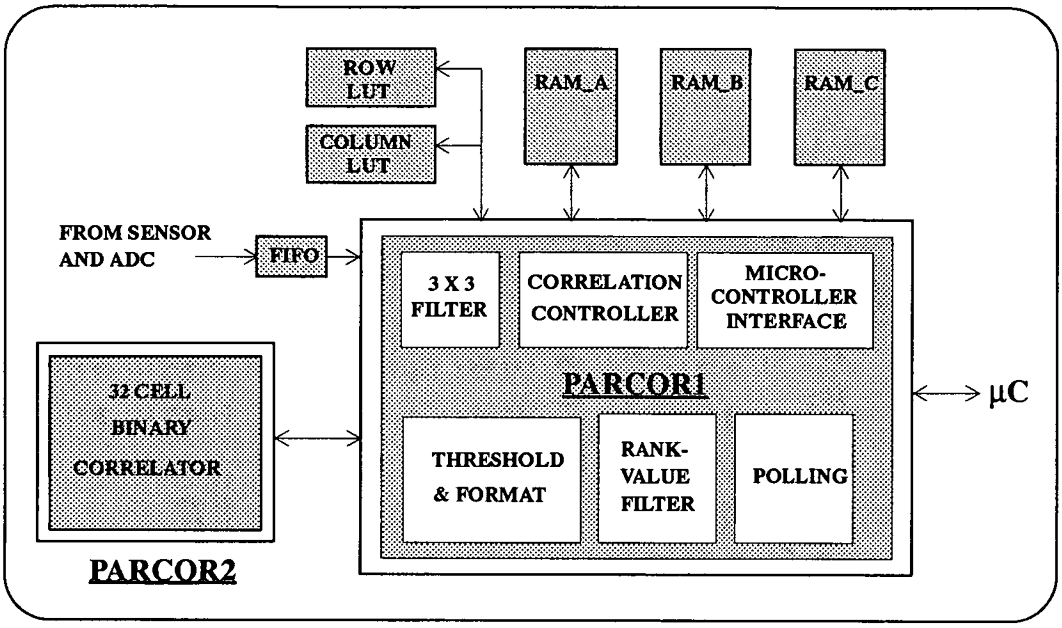


Figure 4.7 Fingerprint Image Processing Sub-system

The off-chip memory provides temporary buffering for the image data at various stages of processing and a store for the selected reference signature. Pre-calculated address modifiers are stored in look-up-table ROMs for use during the correlation process. Each of the autonomous process modules on Parcor1 are independently executable under the control of high-level software running on the FVU systems' embedded 8-bit microcontroller.

Data transfer to, and from, the fingerprint processing subsystem and the microcontroller is achieved *via* an interface situated on Parcor1. As well as access to data and control registers associated with the process modules on Parcor1, the interface allows read access to all associated memory space, *i.e.* RAM, ROM and Parcor2. This feature, combined with the independent control of individual processes, is particularly useful for system testing. Test data can be set up for any part of the process cycle to verify each part of the subsystem. The remainder of this section will review the architectures used to implement each of the required functions.

4.4.2 Microcontroller Interface

Communication and control between the process modules on Parcor1 and the microcontroller takes place *via* on-chip interface logic. The interface takes the form of a set of read-only and write-only registers which are used to control and monitor the operation of the processes performed by the chipset.

An address byte is used to select one of 16 read-only registers and 16 write-only registers. Table 4.3 provides a summary of the registers and their functions. A 5-bit address from the microcontroller is used to select a particular register. The upper bit is effectively a read/write select bit with the four least significant bits selecting a particular register for access. Physically, the interface consists of a 8-bit bidirectional address/data bus plus two timing strobes. The timing of the transfer of an address word to Parcor1 is controlled by the address strobe, *as*. The data timing, to or from the ASIC, is then controlled by a data strobe, *ds*. The registers are of the following types:

- **Write Registers**: Used to set up process parameters, control execution of processes, and modify off-chip memory.

- **Read Registers**: Used to monitor the process status of Parcor1 and access result registers, and view external memory locations (RAM, ROM, Parcor2).

<table>
<tr><th colspan="4">READ ONLY REGISTERS</th></tr>
<tr><th>ADDR</th><th>REGISTER</th><th>FUNCTION</th><th>PROCESS</th></tr>
<tr><td>0</td><td>STATUS</td><td>Shows ASIC operational status</td><td>All</td></tr>
<tr><td>1</td><td></td><td>not used</td><td></td></tr>
<tr><td>2</td><td>THRESH.</td><td>For image monitoring</td><td>Image Capture</td></tr>
<tr><td>3</td><td>PSCORE</td><td>Score returned by polling process</td><td>Polling</td></tr>
<tr><td>4-14</td><td></td><td>not used</td><td></td></tr>
<tr><td>15</td><td>READDATA</td><td>Memory read data register. Low byte</td><td>Memory access</td></tr>
<tr><th colspan="4">WRITE ONLY REGISTERS</th></tr>
<tr><th>ADDR</th><th>REGISTER</th><th>FUNCTION</th><th>PROCESS</th></tr>
<tr><td>16</td><td>CONTROL</td><td>Select and execute a particular process</td><td>All</td></tr>
<tr><td>17</td><td></td><td>not used</td><td></td></tr>
<tr><td>18</td><td>SIGSELECT</td><td>Select signature or signature bank</td><td>Correlation or ranking</td></tr>
<tr><td>19</td><td>ROWCOL</td><td>Select correlation block address</td><td>Correlation</td></tr>
<tr><td>20</td><td>BINTARGET</td><td>Target threshold value</td><td>Image capture</td></tr>
<tr><td>21</td><td></td><td>not used</td><td></td></tr>
<tr><td>22</td><td>RANKTHRESH</td><td>Threshold of interest for correlation results</td><td>Ranking</td></tr>
<tr><td>23</td><td>ANGLES</td><td>Set number of angles for correlation</td><td>Corrank</td></tr>
<tr><td>24</td><td>BEST</td><td>Select number of scores for polling</td><td>Polling</td></tr>
<tr><td>25</td><td></td><td>not used</td><td></td></tr>
<tr><td>26</td><td>ADDLO</td><td>Memory address register. Low byte.</td><td>Memory access</td></tr>
<tr><td>27</td><td>ADDHI</td><td>Memory address register. High byte.</td><td>Memory access</td></tr>
<tr><td>28</td><td>CELLADDR</td><td>Cell address for ranking process</td><td>Ranking</td></tr>
<tr><td>29</td><td>XADDR</td><td>Correlation score position address register. Column</td><td>Polling</td></tr>
<tr><td>30</td><td>YADDR</td><td>Correlation score position address register. Row</td><td>Polling</td></tr>
<tr><td>31</td><td>WRITEDATA</td><td>Memory write data register</td><td>Memory access</td></tr>
</table>

Table 4.3 ASIC Register Map

- **Control Register:** By writing to this register the individual processes can be selected and started. Bit 4 is used to enable ASIC memory access by the microcontroller. This should be asserted before any use is made of the memory access registers

- **Status Register:** The status register is used to determine when a particular ASIC process has been completed. The status register is automatically cleared when a process is started, and the appropriate bit is set in the register when the process has finished. Bit 7 is used to indicate that a new patch threshold has been calculated and is ready to be accessed by the microcontroller.

Memory Access Registers: Two registers, *addhigh* and *addlow*, allow an address within the ASIC memory map to be selected. Before writing to either of the address setup registers, or the *writedata* register, the memory access bit within the *control* register must be set. After completion of the required memory accesses, this bit must be reset by writing zero to the *control* register. The action of setting up an address, by writing to *addhigh* or *addlow*, causes a read operation from the specified location which is then stored in the *readdata* register. This register can then be read by the microcontroller. When reading the cell array only the lower 6 bits of the result in *readdata* are valid. To write to a specific memory location an address is setup by writing to *addhigh*, then *addlow*. When the required data is then written to the *writedata* register, the selected memory location is accessed and updated with the new value. Note that the action of setting up the address causes a read operation to take place, so that the original contents of the ASIC memory is available until the next memory access is performed.

4.4.3 Smoothing Filter and Thresholding

These processes form a binary thresholded version of a 256 x 256 x 7 bit image. They do so 'on the fly' so that live fingerprint image data are consumed from the sensor subsystem to form a complete binary image which is left in a 64kbit RAM. The early image is incidentally smoothed *via* a two-dimensional convolution operation, prior to binarization. The data flow for the smoothing and thresholding processes are shown in figure 4.8.

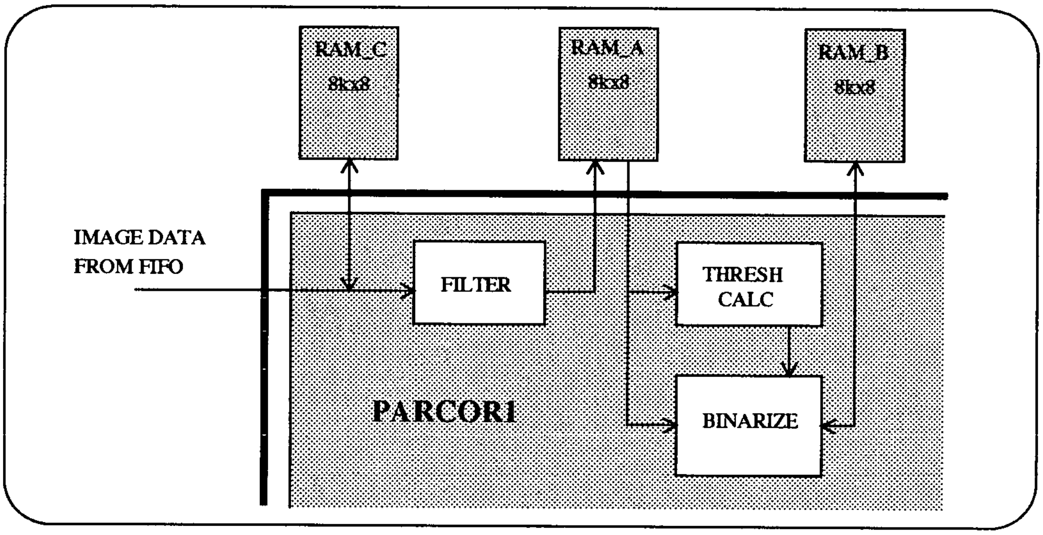


Figure 4.8 Image Preprocessing Dataflow

The process architecture has been carefully arranged so that there is no requirement to buffer the full grey-scale image. Memory utilization for the preprocessing functions is as follows:

- **RAM_C**: Dynamically buffers the previous two image lines, as required by the smoothing function.
- **RAM_A**: Acts as a flip-flop buffer for the smoothed grey-scale data. While one half (4kbytes) is being processed by the thresholding function, new filtered data is written to the other.
- **RAM_B**: Buffers the full 64kbit binary image formatted ready for the next stage of processing, binary correlation.

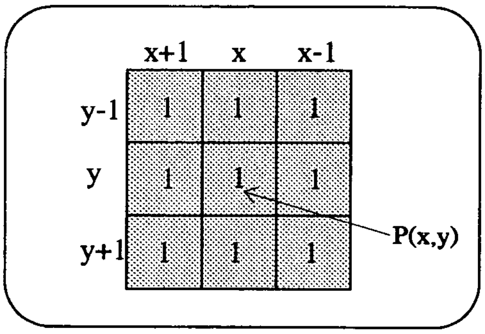


Figure 4.9 3x3 Convolution Window

The grey scale image is first smoothed using a 3 x 3 convolution window with unity weights as shown in figure 4.9. The result is a local average of the centre pixel with its eight nearest neighbours. If the pixels are indexed by column and row as P(x,y) then the smoothing operation is:

$$P_{ave}(x,y) = \frac{\sum_{x=-1}^{1}\left\{\sum_{y=-1}^{1} P(x,y)\right\}}{8}$$

Note: edge-effects affect only the border pixels, and are ignored (*i.e.* they may be computed erroneously, if necessary).

The architecture for this filter operator is given in figure 4.10. The two-line delay is realized in the off-chip memory, RAM_C. The input to the filter is 7-bit grey scale image data sourced either from the FIFO or RAM_C. The datapath consists of two multiplexed adders. The first adds columns of 3 pixels. The second adds the results of column summations. The filter operator is fully pipelined, so new data is read continuously. Once the datapath pipeline is fully loaded it produces one filtered pixel value every three system clock cycles.

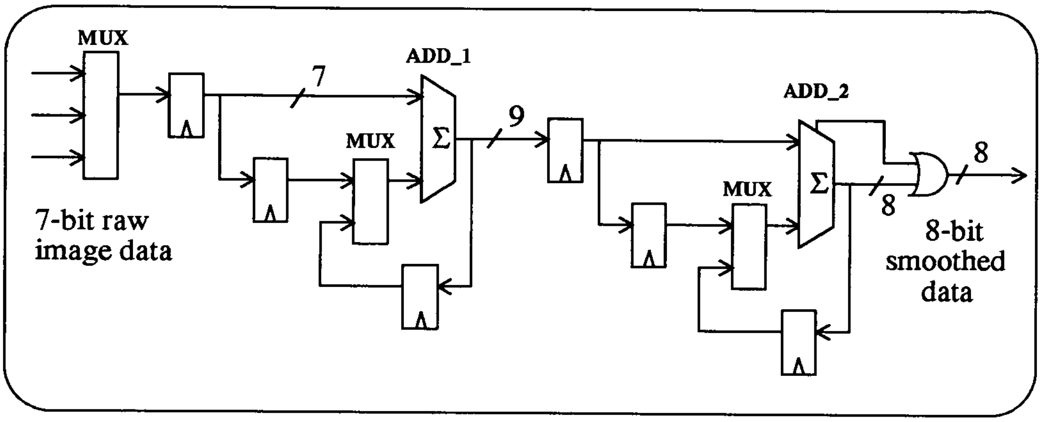


Figure 4.10 Smoothing Filter Datapath

The initial 7-bit data grows to 8-bits after the first pixel addition, then to 9-bits at the second. At the first column addition they grow to 10-bits, and finally to 11-bits at the second. The maximum possible signal value at this point is 9 x 127 = 1143, which is only marginally above the 10-bit range. An 8-bit result is recovered by first dropping the least significant 2-bits (equivalent to dividing by 4). Then, if the top bit is set (which indicates overflow), the remaining 8-bits are hard clamped to 255.

The filtered grey-scale data is buffered in off chip RAM until there is sufficient to commence the binary thresholding. This involves taking the filtered grey-scale data and producing a normalized black and white representation of the ridge/trough pattern of the fingerprint image.

Patching is the process of splitting the filtered grey-scale image into 256 patches of 16 x 16 pixels. For each patch a threshold is calculated (*thresh_calc*), and then applied to the patch data categorizing each pixel as either black or white (*binarize*). A 'target' for the number of white pixels required (usually 50% but 25% or 75% during signature generation) is set and then an optimum threshold, guaranteed to give at least the target score, is calculated by the following procedure.

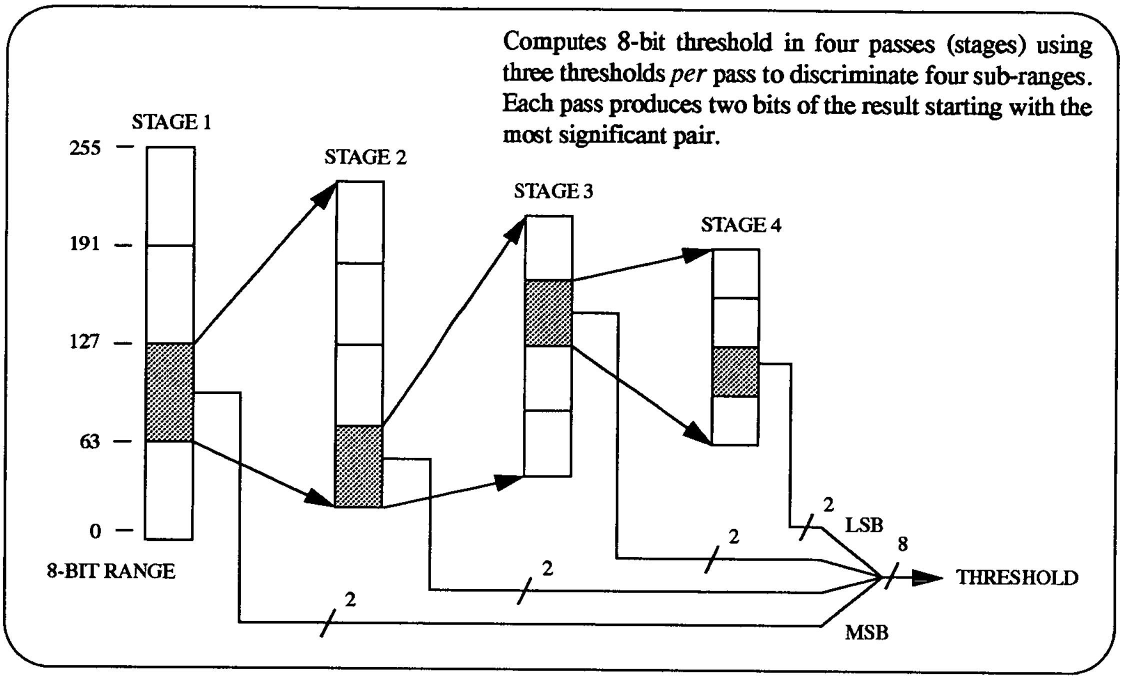

Computes 8-bit threshold in four passes (stages) using three thresholds *per* pass to discriminate four sub-ranges. Each pass produces two bits of the result starting with the most significant pair.
STAGE 1
STAGE 2
STAGE 3
STAGE 4
255
191
127
63
0
8-BIT RANGE
2
2
2
2
LSB
8
THRESHOLD
MSB

Figure 4.11 Successive Approximation Process

The threshold calculation process (see figure 4.11) is based upon a successive approximation technique whereby a threshold is estimated, and then refined in binary steps. The computational load is reduced by applying the procedure to only one quarter of the pixels in each patch, by restricting the search to four stages (binary steps), and by using three thresholds *per* pass to discriminate four subranges. The architecture for this process is shown in figure 4.12. This produces an 8-bit threshold, which is then applied to the whole of the patch to produce the binary image with, at least, the target number of white pixels.

The result of the thresholding process is a 64k binary image, which is packed into bytes and stored in RAM_B. Each grey-scale value for a patch is compared with the threshold and the result written to the appropriate bit location, selected by *bsel*, in a byte read from RAM_B.

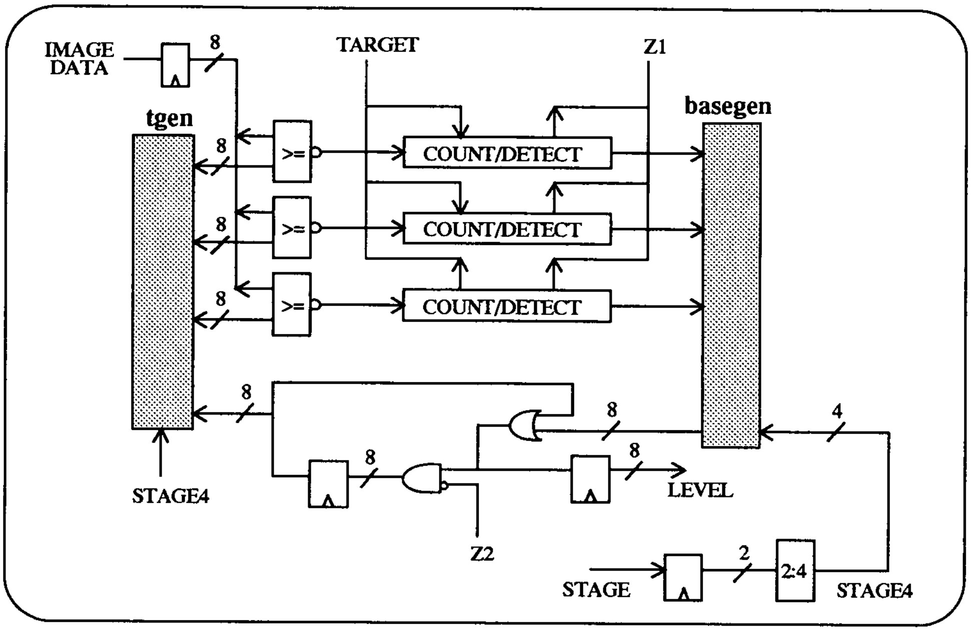


Figure 4.12 Threshold Processor, *thresh_calc*

The format of the stored binary image data, developed by Bruce[22], is such that if the image is divided into 8 horizontal strips of 32 lines, a byte read from RAM_B would contain one bit from the corresponding position in each of the 8 strips (see figure 4.13).

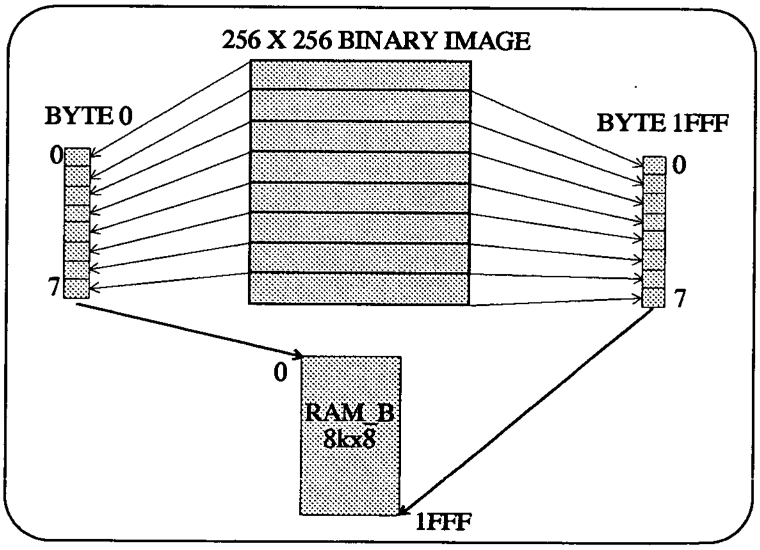


Figure 4.13 Binary Image Storage Format

This format allows fast data access during the computationally intensive correlation phase of the verification algorithm.

4.4.4 Correlation

The comparison process involves a series of binary correlations between a fingerprint image and reference signature templates. Conceptually, the templates are compared with the fingerprint image for various positions and angles of rotation, looking for variations with maximum correlation.

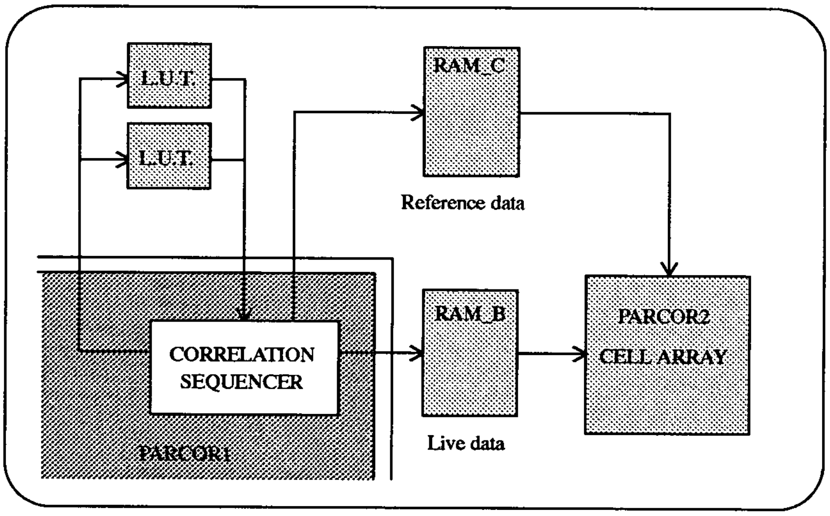


Figure 4.14 Correlation Dataflow

Central to the hardware architecture is a correlation cell which counts coincidences in the pixel streams, generated from the image and signature. The cell retains a best score that is updated, if larger than the previous score, after every 128 point correlation. Parallelism is achieved in two ways. Firstly, the image is divided into 8 horizontal strips and these are correlated simultaneously against the same signature, using a column of 8 cells. Secondly, 4 signatures are correlated simultaneously against the same image data using four rows of cells. Once the 4 signatures have been fully correlated within the predefined limits of position and rotation, the next four signatures are used and so on, until all 16 sub-signatures have been used. This array of 32 correlation cells is implemented as a second ASIC device, Parcor2. An architectural overview of this chip is given in section 4.4.5.

The dataflow diagram for this correlation process is shown in figure 4.14. The comparison process control module, on Parcor1, forms addresses to transfer the correct binary image data and signature data to the cell array for the current position and angle.

The 64k binary image, in RAM_B, is not stored in raster scan format. The 256 x 256 pixel image is split into 8 strips of 32 rows by 256 columns, and organized such that an address refers to a single bit in each 32 x 256 strip. Thus, when the RAM is accessed it returns the eight pixels corresponding to the same location in each strip. The signature data is stored in RAM_C. The data is organized so that when a byte is accessed the data and *don't care* bits for 4 signatures are accessed simultaneously.

To take into account the rotation of the image, pre-computed offsets to modify the address used to access the fingerprint image data are stored in EPROMs (one for row offsets and one for column offsets), which share the address/data bus for RAM_A. These address modifiers are read by the control hardware and combined with the true address to form the virtual address. The result of this combination (either addition or subtraction) results in the generation of the row overflow, row underflow and column error signals used to control the function of the correlation cell array.

4.4.5 Correlation Array

Parcor2 performs the task of correlation which is the core of the fingerprint matching algorithm. Central to the hardware architecture is an array of correlation cells which count the coincidences in data streams generated from the fingerprint image and the signature data (see figure 4.15). Each cell also retains a best score that is updated, if larger than the previous best score, after every 128 point correlation. The 8 x 4 matrix allows 8 image lines to be correlated with 4 signatures simultaneously.

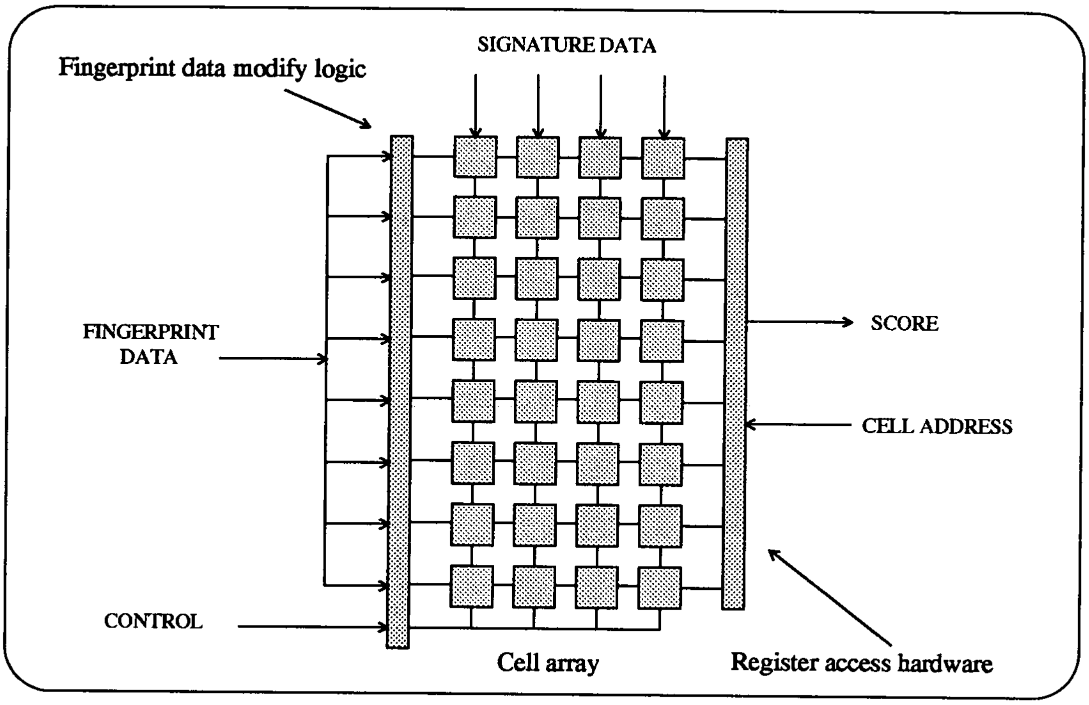


Figure 4.15 Correlation Cell Array

Additional hardware is provided to shift the fingerprint data up or down when error signals are produced by the correlation controller on Parcor1. A 5-bit address and on-chip decoder allows each cell to be accessed and its best score register read.

The matching process is performed using a parallel array of correlation cells. A cell counts the number of matches between the fingerprint under analysis and the reference print for a number of translations and angles. Each cell also maintains a maximum score which is updated after each 128 point correlation.

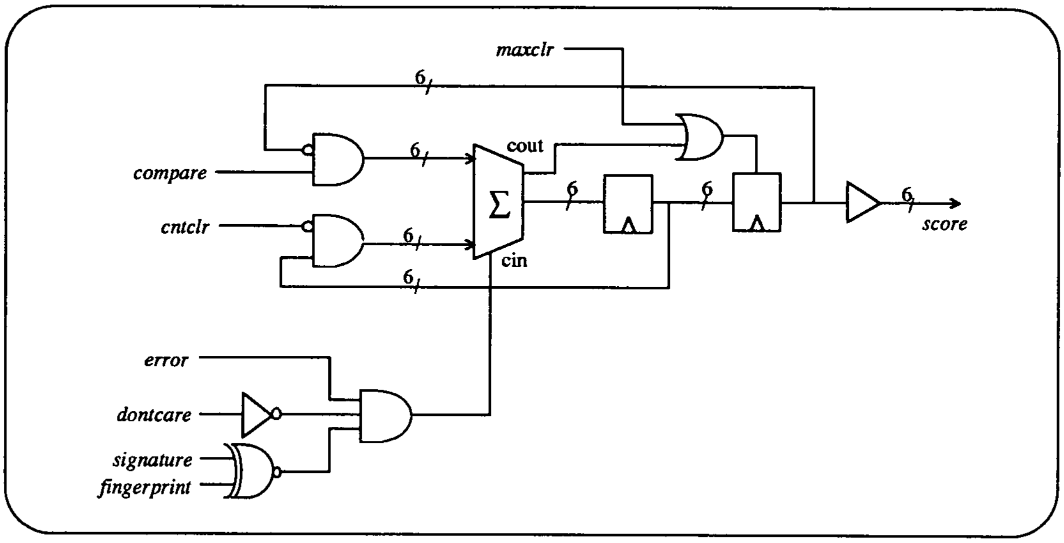


Figure 4.16 Correlation Cell

A correlation cell, figure 4.16, is constructed around a 6-bit ripple adder with carry-in and carry-out flags which counts the number of coincidences between the fingerprint and signature data for a particular position and angle. Although each correlation is 128 points, a maximum score of 63 is set by the use of the *don't care* bits in the signature data. After each 128 point correlation, the result is compared with the current highest score. This is achieved by inverting the highest score and adding it to the new score. If a carry is produced, the new score is latched and held as the current maximum. It is vital that, during the comparison period, the carry-in to the adder is held low. The *error* signal, which is generated from the column error signal, *colcout*, is used to control the carry-in bit. Signals used to control the operation of a correlation cell include *cntclr* (reset the counters), *maxclr* (clear the best score registers), *compare* (force a score comparison), and *cebar* (enable the score output buffers).

After a sequence of 128 point correlations, the process is halted and all the correlation results are read out and ranked in ‘best score’ tables in RAM_A (see section 4.4.6). The array is then reset and the maximum score registers cleared ready for the next correlation sequence to be executed. This correlation-rank process is repeated for each of the 4 banks of 4 sub-signature templates and various positions.

4.4.6 Rank Value Filter

This Parcor1 process reads a correlation score from the correlation cell array (Parcor2) and performs a ranking function on the score. Provided that the magnitude of the score is greater than the ranking threshold level, *rankthresh*, the score is dynamically ranked in a table of the top 16 scores for the current signature. A similar table is maintained for each of the 16 signatures.

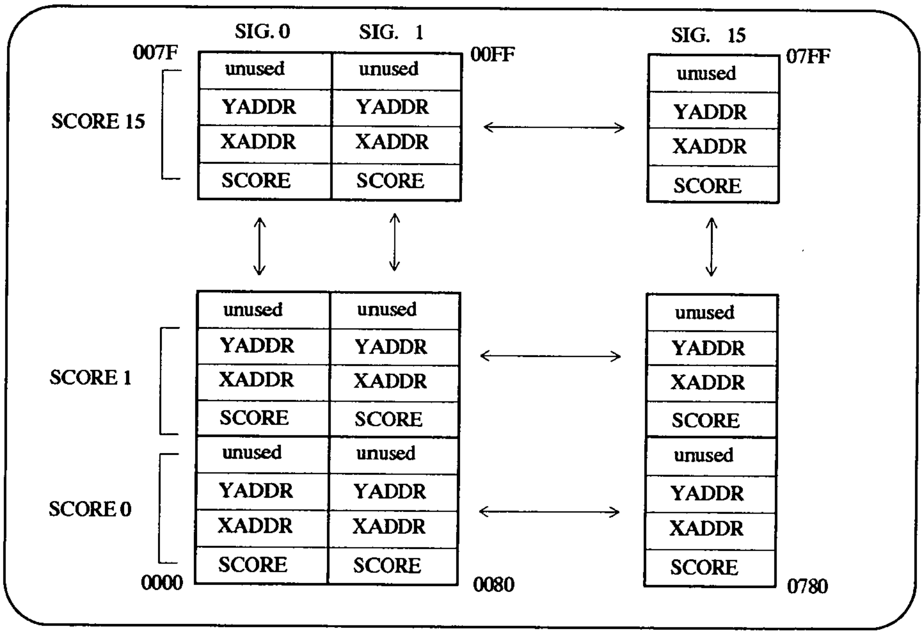


Figure 4.17 Rank Table Format

The data stored at each location within a rank-table, figure 4.17, consists of 3 bytes, the score, and a 2-byte address relating each score to a position in the 32 x 32 block grid covering the fingerprint image space. When a new score is to be ranked the data corresponding to a particular rank-table is read from memory, starting with the highest ranked score. The new score is compared with each successively lower score until it is found to be equal or lower in magnitude. It is then slotted into the rank table along with its associated address bytes. This causes the previous lowest score to drop-off the bottom of the table. If the new score proves to be lower than the any of the existing ranked scores no changes are made to the table and the new score is discarded.

The ranking process datapath, figure 4.18, is constructed around two 6-bit wide, 3-word deep FIFOs which provide on-chip buffering for scores and their associated address bytes (*xaddr, yaddr*). Multiplexers are used to direct the flow of data around the module. The signals used to control the dataflow (*c1 - c6*) are a function of fixed process cycle timing and data comparisons and are generated in a local datapath sequencer.

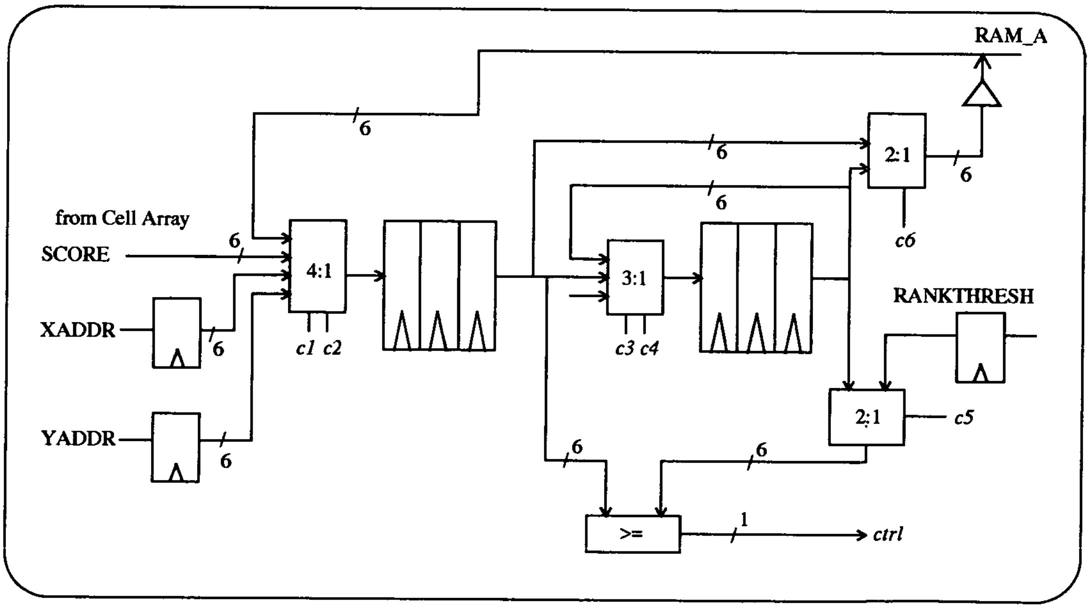


Figure 4.18 Rank Value Filter Datapath

4.4.7 Polling

The polling function is the final stage of the comparison process. The results tabulated during the correlation and rank procedures are analysed to determine the level of similarity between the fingerprint image and the reference templates. To accelerate this process, a simple module is provided which searches the best score tables in RAM_A, counting the number of sub-signatures with ranked scores within a specified 2 x 2 area of the 32 x 32 block grid covering the fingerprint image space. Hence if the specified address is {x,y} it also searches for scores with addresses {x,y+1}, {x+1,y}, {x+1,y+1}. If a score is found in the tables with a valid address for a particular sub-signature a count is incremented. This gives a a maximum possible poll score of 16 (assuming a full search of the best score tables). The datapath schematic for this function is given in figure 4.19.

Assuming a sufficient number of sub-signatures for a given area of the image have ranked scores, the result is a positive match between the presented image and the reference fingerprint.

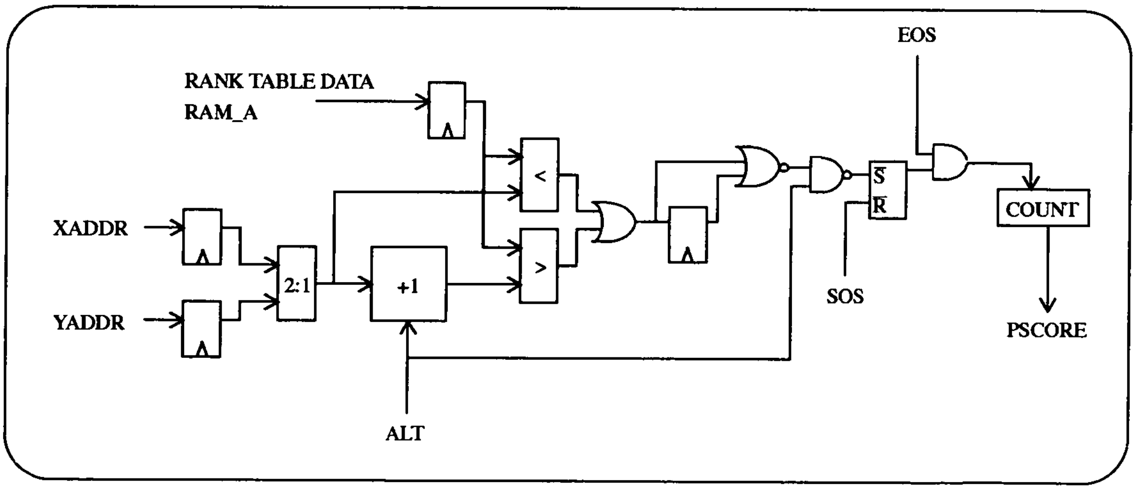


Figure 4.19 Polling Operator Datapath

4.5 ASIC IMPLEMENTATION

The architectures described in the previous sections were implemented as two digital ASICs, Parcor1 and Parcor2, using silicon compilation design techniques. Both devices were fabricated by European Silicon Structures (ES2), but were implemented using different software tools. This section will briefly describe the design method followed for each device. A summary of the main physical characteristics for Parcor1 and Parcor2 are presented in table 4.4

CHARACTERISTIC	PARCOR1	PARCOR2
Foundry	ES2	ES2
Process	2μm	2μm
No. of transistors	43,870	20,884
Die width	9.03mm	8.49mm
Die height	9.23mm	7.62mm
Die area	83.32mm^2	64.62mm^2
Clock frequency (typ.)	12MHz	12MHz
Clock frequency (max.)	13MHz	20MHz
No. of I/O pins	84	45
Power (@5V & 12MHz)	320mW	196mW
Package	84 pin J-lead	68 pin J-lead

Table 4.4 FVU ASIC Physical Characteristics

4.5.1 Parcor1

Parcor1, the image processing ASIC, was designed using the SOLO-1400 silicon compiler[95] which provides a fast route for the implementation of digital ASICs for fabrication by ES2 and their second sources. Figure 4.20 shows the design flow for SOLO-1400 from design entry, through simulation and physical layout, to design validation prior to shipment to ES2 for fabrication.

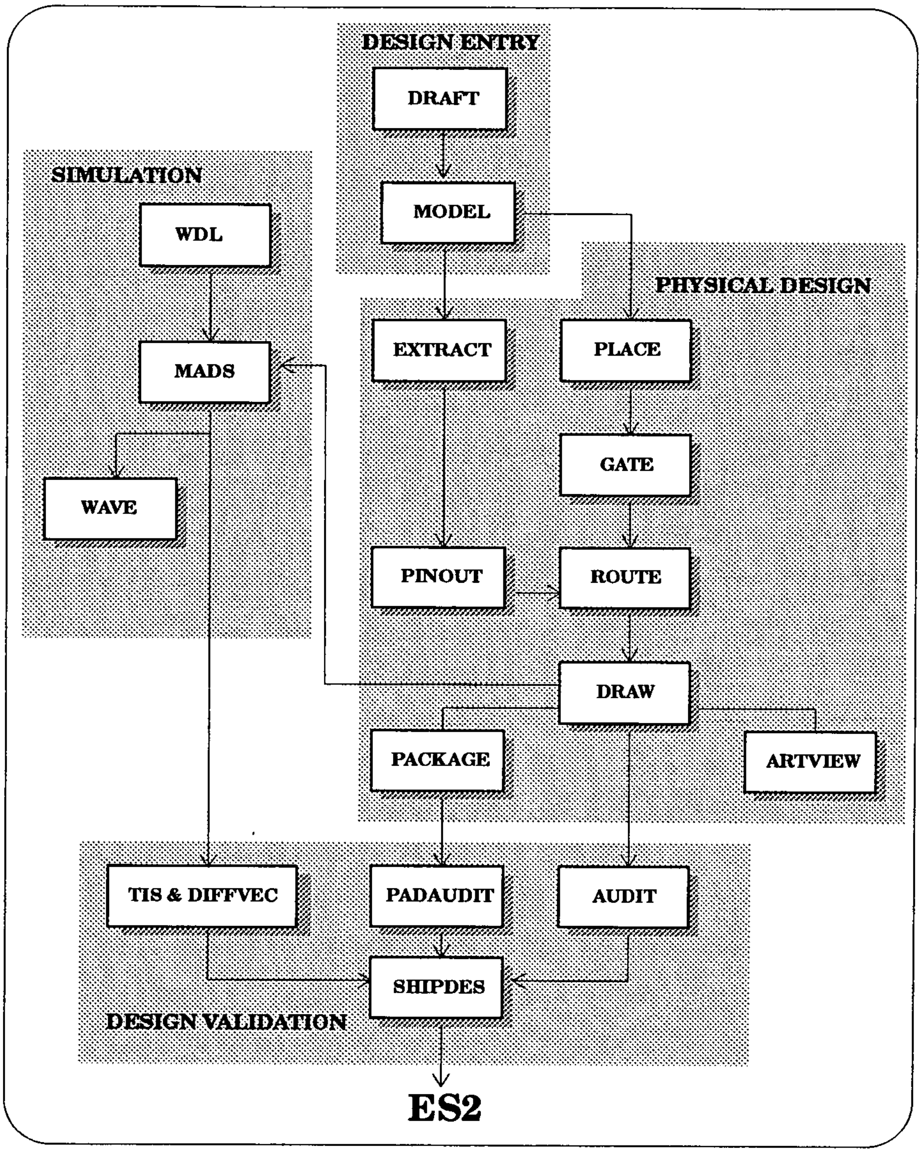


Figure 4.20 SOLO-1400 Design Flow

The circuit is entered either as a logic diagram *via* a schematic-capture package or in text form using a proprietary hardware description language (HDL) called *model*. In this instance *model* was used since this made it easier to describe, in a compact manner, parallel architectures such as the correlation cell array. The design is built up hierarchically. At the lowest level are basic design elements (logic gates, latches, counters) selected from ES2 part libraries. These parts are then grouped together to form the higher level operators required to implement the architecture. Once the design description is captured and successfully compiled, further software tools are used to automatically generate design models for simulation and the physical layout of the ASIC.

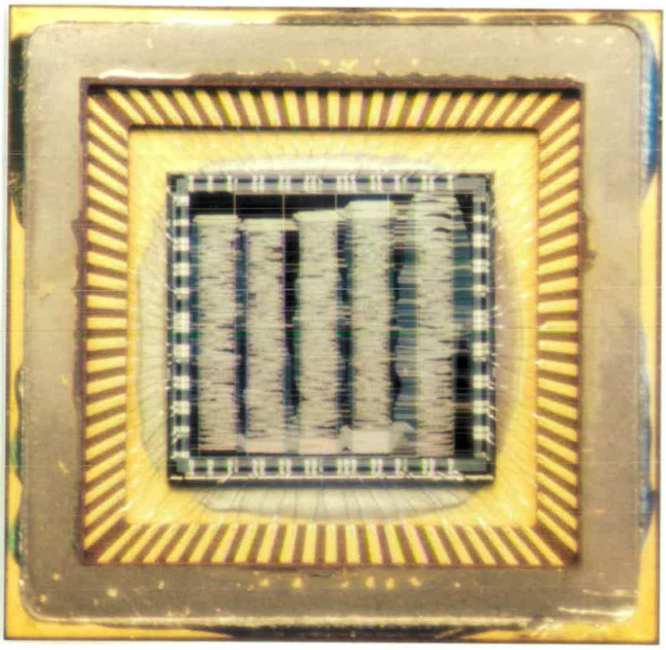


Figure 4.21 Photograph of Parcor1

The floorplan structure of a SOLO generated design consists of a small number of columns, containing rows of CMOS transistor pairs (one p-type and one n-type device) known as stages. Every library part and user defined block is implemented as a series of interconnected stages. Redundant logic is automatically removed from the design by the compiler before the layout process takes place. Functional simulation was performed both prior to, and after, layout when more accurate timing and loading information was available

After performing post layout simulation and design verification, the artwork and simulation test vectors for the circuit are taped-out and shipped to ES2 for the fabrication of prototype devices. Parcor1 (figure 4.21) contains all the logic, except the parallel correlation array, required to implement the fingerprint image preprocessing and comparison algorithm.

4.5.2 Parcor2

The second ASIC, Parcor2, implements the parallel correlation array which accelerates the computationally intensive part of the fingerprint comparison process. This device was implemented using the Genesil[104],[94] suite of integrated circuit design tools. Unlike SOLO-1400, which is linked to a single silicon foundry, Genesil allows a design to be captured then compiled for fabrication by a variety of silicon vendors (including ES2).

The design capture method involves the selection of the required building blocks from the on-line library of parameterised building blocks such as memory, registers, arithmetic modules, input/output structures, test elements and basic logic gates. On-screen ‘forms’ are completed to define the parameterisable elements for each instance of a block. These blocks are then grouped to form higher level modules which, in turn, are connected together to complete the design. Once the functional description is captured the software automatically generates internal models for logical function, timing, power and layout. A graphical view of each module is also generated at this stage. Functional simulation then provides verification of the logical operation of the captured circuit against its design specification. Functional simulation may be performed at any level in the design hierarchy (block, module, chip, system). Static timing analysis might also be performed at this stage. The Genesil timing analyser exhaustively examines every possible circuit path, calculating delays for various process parameters, operating temperatures, and supply voltages. The results are automatically analysed to provide values for the maximum operating frequency, and a list of the critical paths for each module.

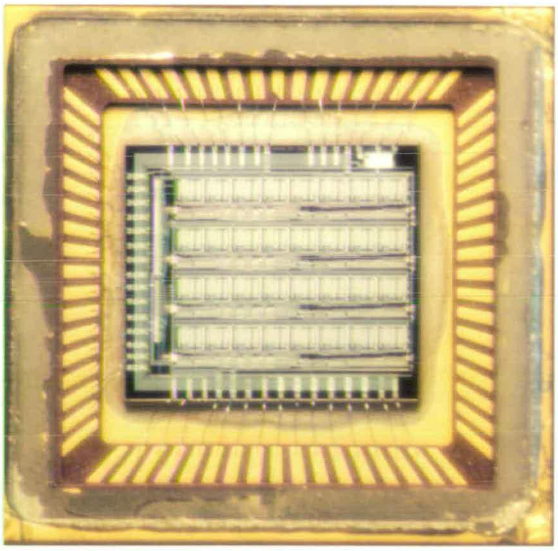


Figure 4.22 Photograph of Parcor2

After functional and timing verification at the module level has been completed the design is ready for floorplanning. During this phase of the design flow, the small number of compiled modules (including I/O pads) forming the complete chip are interactively placed to produce an efficient floorplan. The software then automatically routes the power, control, and data lines between the blocks and the I/O pads. The design is then ready for final simulation and timing analysis to verify functionality and performance. Once completed, the design artwork and test vectors are taped out and sent for fabrication. Parcor2 was fabricated using ES2's 2.0μm CMOS process. This chip was the first device to be fabricated using this combination of silicon design software and silicon vendor. A photograph of the resulting device is given in figure 4.22. Note the regular structure due to the datapath architecture adopted for this design.

4.6 DISCUSSION AND CONCLUSIONS

The FVU system has been developed based around a custom image processing ASIC chipset. An alternative implementation approach would have been to utilise standard function specific image processing devices or powerful signal processing microprocessors. A major advantage of this approach would have been the significantly reduced engineering costs involved in developing the FVU system.

Set against this advantage is that standard parts either do not have enough processing bandwidth or architectural flexibility implement all of the required functions. The fingerprint comparison algorithm consists of a number of different functions each with different processing requirements. For example any standard part capable of implementing the low level parallel processing correlation part of the algorithm would not be an appropriate device to perform any of the higher level data analysis functions.

The support memory for the ASIC implementation is very efficiently utilised since it was organised to suit the dataflow requirements of the algorithm rather than those of a particular standard device. Another potential advantage of implementing the fingerprint system in ASIC form is algorithmic security. It is much more difficult to copy an ASIC than it is to replicate software.

The use of VLSI technology in the development of the prototype fingerprint access control system has produced a compact, powerful, real-time image processing system. Further size, cost, and performance gains can be achieved by applying the leverage of VLSI to elements of the system beyond the core digital processing elements. The next chapter will describe the design of an architecture for a chip which efficiently combines the whole of the image sensing, preprocessing functions and comparison algorithm into a single highly integrated device.

Chapter 5. IMAGE SENSOR-PROCESSOR

5.1 MACRO-ARCHITECTURAL SPECIFICATION

This chapter describes the architectures developed to implement the fingerprint sensing, capture, and comparison system as a single, highly integrated VLSI device, ASIP (Application Specific Image Processor). Figure 5.1 shows the functional schematic of the prototype FVU system, as described in chapter 4. The single chip sensor-processor replaces the sensing, image capture, preprocessing, and image comparison functions as indicated by the shaded area.

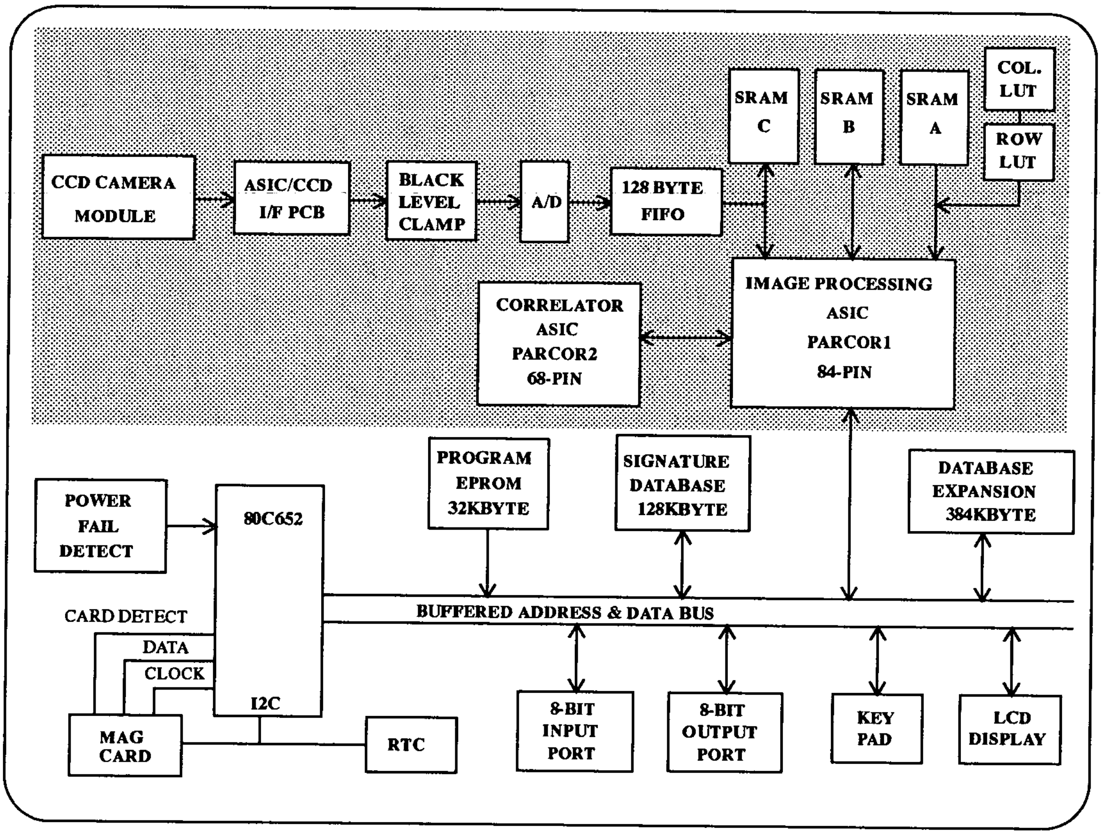


Figure 5.1 FVU Functional Schematic

The principal functional modules forming the fingerprint comparison sub-system are the same as the prototype FVU, but the architectures and design methods used to implement the individual parts of the algorithm are quite different. A detailed description of the architectural design of the various elements of the integrated image sensor-processor will be presented in this chapter. The benefits of this design approach will be highlighted using the single board, prototype FVU system as a benchmark.

Figure 5.2 shows the main functional elements of the integrated sensor-processor. Compared with the existing system architecture, the most radical change is the integration of the image sensor with the digital image processor on the same substrate. This has been made possible by the recent development[85],[87] of a high quality, customisable sensor array designed to be fabricated using the same standard low-cost ASIC CMOS process as is used to fabricated the surrounding digital logic. This technology will be described in more detail in section 5.2.

The external memory requirements have been reduced from 192kbit to 64kbit by including on-chip, fast access, SRAM tailored to the needs of the individual processes. This is achieved through the use of compilers which can generate memory blocks of any required size (*i.e.* word length x number of words). A single, off-chip, 64kbit SRAM has been retained to buffer the binary image produced by the image capture and preprocessing function. The decision not to integrate this memory was based on the economics of integrating large RAMs.

Image capture and preprocessing (digitisation, filtering and binary thresholding) are performed on-chip without the need for interaction with the high level control software running on the microcontroller. To increase the performance of the comparison process, the number of correlation cells has been increased from 32 to 64. This change to the architecture alone reduces the fingerprint comparison time by 50%, from 1.5 seconds (typical) to 0.75 seconds (typical).

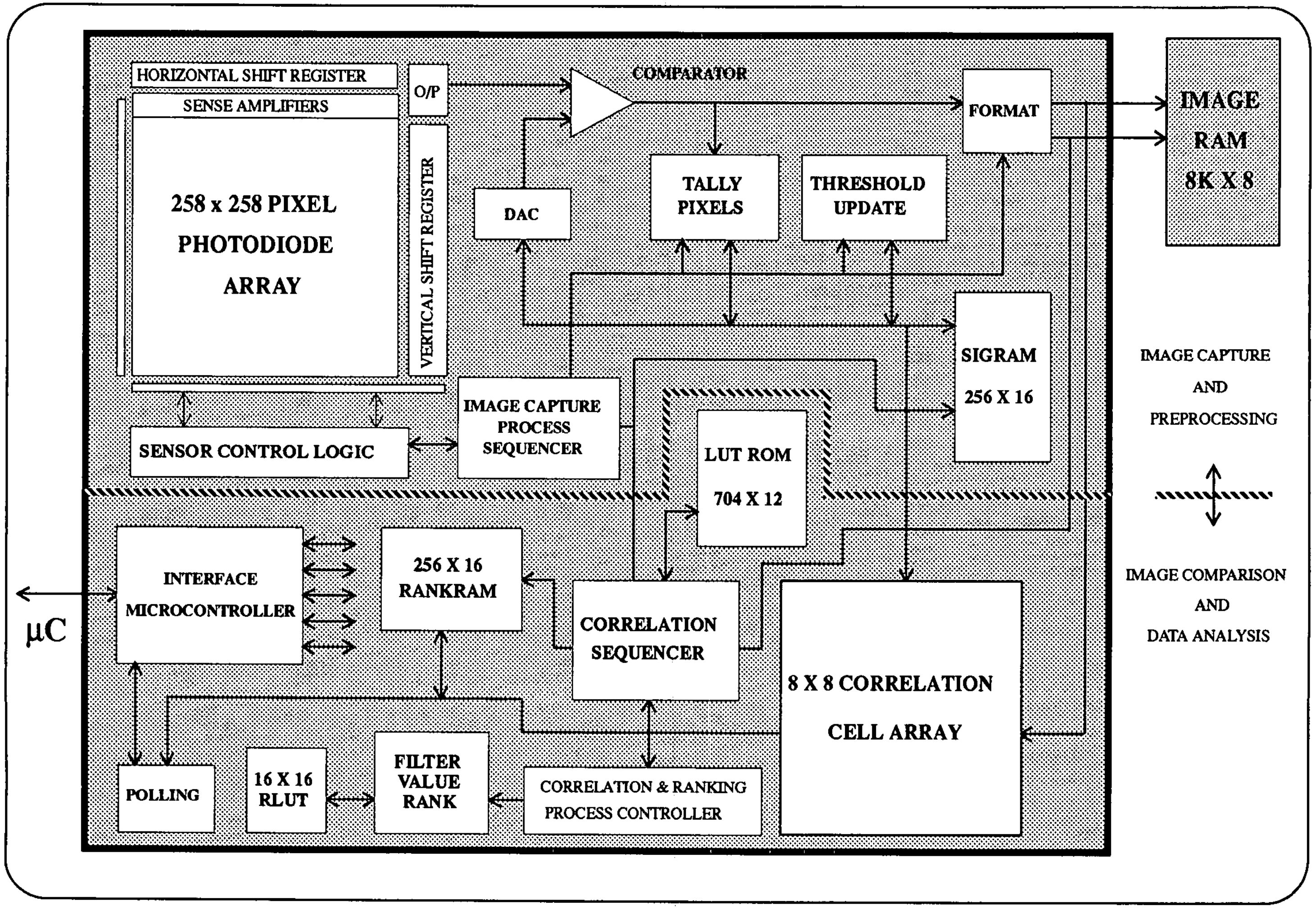

HORIZONTAL SHIFT REGISTER
SENSE AMPLIFIERS
258 x 258 PIXEL PHOTODIODE ARRAY
O/P
VERTICAL SHIFT REGISTER
COMPARATOR
DAC
TALLY PIXELS
THRESHOLD UPDATE
FORMAT
IMAGE RAM 8K X 8
SIGRAM 256 X 16
SENSOR CONTROL LOGIC
IMAGE CAPTURE PROCESS SEQUENCER
LUT ROM 704 X 12
IMAGE CAPTURE AND PREPROCESSING
IMAGE COMPARISON AND DATA ANALYSIS
μC
INTERFACE MICROCONTROLLER
256 X 16 RANKRAM
CORRELATION SEQUENCER
8 X 8 CORRELATION CELL ARRAY
POLLING
16 X 16 RLUT
FILTER VALUE RANK
CORRELATION & RANKING PROCESS CONTROLLER

Figure 5.2 ASIP Functional Schematic

The address modifiers used during correlation, to rotate the image with respect to the reference templates, are now stored in on-chip LUT ROM. Autonomous execution of the correlation and ranking processes has been provided, easing the computational load on the microcontroller and reducing the execution time. The ranking process, which reads the high scores from the correlation cell array, has been modified reducing the memory requirements for the best score tables and increasing its throughput.

READ ONLY REGISTERS			
ADDR	**REGISTER**	**FUNCTION**	**PROCESS**
0	STATUS	Shows ASIC operational status	All
1 - 2		not used	
3	PSCORE	Score returned by polling process	Polling
4 - 13		not used	
14	READHI	Memory read data register. High byte	Memory access
15	READLO	Memory read data register. Low byte	Memory access
WRITE ONLY REGISTERS			
ADDR	**REGISTER**	**FUNCTION**	**PROCESS**
16	CONTROL	Select and execute a particular process	All
17	TARGET1	Tally target value (highest)	Image capture
18	SIGBANK	Select one from two banks of 4 signatures	Corrank
19	ROWCOL	Set sensor exposure or select correlation block	Image capture or corrank
20	TARGET2	Tally target value (middle)	Image capture
21	CORRTEST	Set number of 128 point correlations	Corrank
22	TARGET3	Tally target value (lowest)	Image capture
23	ANGLES	Set number of angles for correlation	Corrank
24	BEST	Select number of scores for polling or DAC test input	Image capture or polling
25	SENTEST	Sensor, DAC and comparator test select register	Image capture
26	ADDLO	Memory address register. Low byte.	Memory access
27	ADDHI	Memory address register. High byte.	Memory access
28	XADDR	Correlation score position address register. Column	Polling
29	YADDR	Correlation score position address register. Row	Polling
30	WRITEHI	Memory write data register. High byte	Memory access
31	WRITELO	Memory write data register. Low byte	Memory access

Table 5.1 ASIP Register Map

The final analysis of the best score tables, to determine whether or not a fingerprint and the selected reference match, is still performed by software, running on the 8-bit microcontroller. Like Parcor1, a hardware module has been provided on-chip to accelerate this process by scanning the best score tables and counting coincidences of ranked scores with particular block addresses. This information is then used by the control software to determine the quality of match between the image and the reference. The microcontroller interface is very similar to that used in Parcor1, although the register map has changed to accommodate the requirements of the new architectures. Table 5.1 provides a summary of the registers used in ASIP and a brief description of their functions.

The remainder of this chapter looks in detail at the architectures used to implement the fingerprint matching algorithm, highlighting the various novel approaches used to produce ASIP. First, however, it is necessary to briefly review the CMOS image sensing technology that has enabled this device to be designed.

5.2 CMOS IMAGE SENSORS

There is a wide range of commercially available solid state image sensors based on a variety of technologies all of which are suitable for image processing applications. Unfortunately, none of these devices are manufactured using the same low-cost ASIC CMOS process which is used to implement the image processing logic. Research at Edinburgh University [84],[75],[85],[87],[91] has resulted in the development of a novel photodiode based, image sensing technology, fabricated using a standard digital CMOS process.

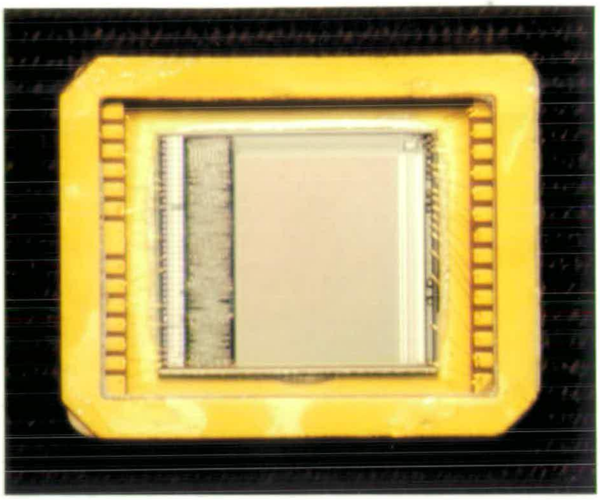


Figure 5.3 Typical CMOS Image Sensor

The performance of this technology matches that of CCD (the most commonly used image sensor technology) in almost every area *e.g.* linearity, dark current, spectral sensitivity, signal to noise ratio, blooming protection. Compared to CCD based cameras, the Edinburgh CMOS sensor has the added advantages that it only needs a single 5V power supply, and a simple single-phase clock. Additionally, the control and signal formatting logic can be on the same silicon substrate as the sensor array. Further features of the technology which are particularly interesting from a system designer's point of view include:

- Customisable pixel pitch.
- Customisable array size.
- Portability to different CMOS processes.
- Electronic exposure control.
- Can be integrated with custom processing logic.

The sensor is based on an array of MOS devices used to form a photodiode, one for each pixel. A photodiode is formed by extending the source region of the MOS device (see figure 5.4).

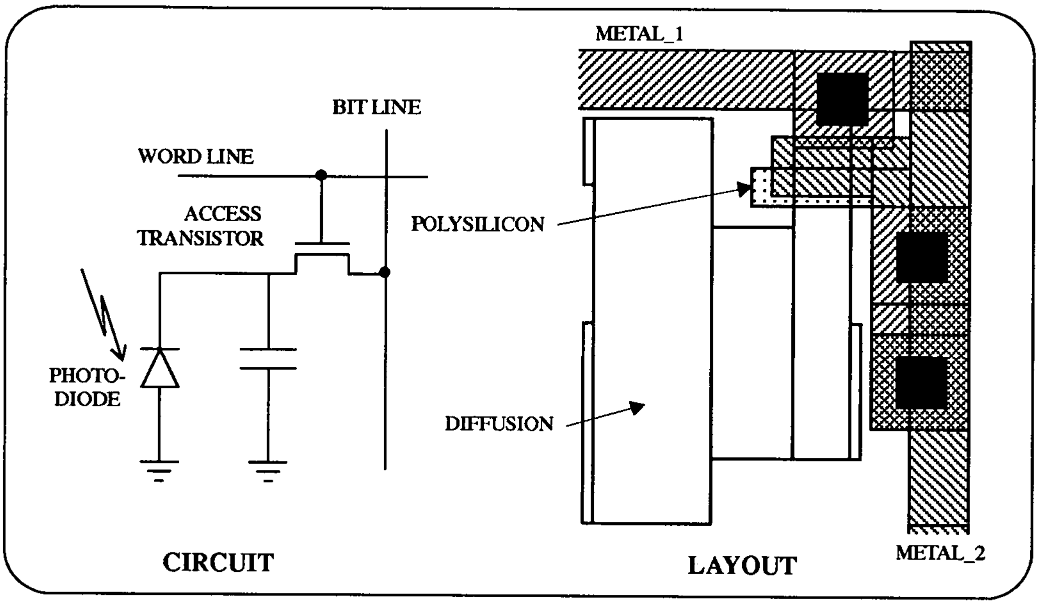


Figure 5.4 MOS Photodiode

A row of photodiodes are charged then isolated *via* a common word line connecting the gate of each MOS transistor. Light falling on the source region of a pixel produces a small photo current which gradually discharges the device. After a short integration period, a pixel is read by selecting the transistor and dumping the remaining charge onto a bit line. Each bit line connects a column of pixels together. Pixels in any given column are accessed individually by enabling a row at a time. At the top of each bit line is a sense amplifier pitch matched to the pixel column width (typically 10-20μm). this converts the charge packet into a voltage which is sampled and stored on a capacitor.

The design of this sense amplifier is critical to the near CCD performance of this

MOS image sensor. The stored values can then be sequentially accessed and output *via* a final amplification stage. The timing of the activation of the word lines and pixel columns are controlled by horizontal and vertical digital shift registers, each pitch matched to the respective pixel dimension. All other sensor timing and output formatting signals are generated by digital control logic, implemented on the same silicon substrate as the sensor array. This module can also be expanded to include application specific functions such as video formatting and automatic exposure control.

Circuitry is also provided that allows the post-fabrication testing of the sensor to be performed using conventional digital-logic test techniques. Basically, predetermined digital data is loaded into the bit-lines, word-lines, and pixels and the output patterns compared with the expected results. A full description of this test strategy is available elsewhere[90].

Finally, the flexibility to produce a image sensor tuned to the requirements of the fingerprint verification algorithm at a cost, in production volumes, of substantially less than that of a standard CCD sensor, makes a CMOS based sensor particularly suitable for this application.

5.3 IMAGE SENSING AND PRE-PROCESSING

To illustrate the advantages offered by integrated sensor-processor systems, this section looks in detail at the image sensing, digitisation, smoothing and normalisation functions of the fingerprint matching system.

5.3.1 Architectural Overview

In the prototype FVU system the image sensing and preprocessing functions were performed in the following manner. First the analogue signal from the CCD sensor module was digitised using a video-rate, analogue-to-digital convertor (ADC) to give a 256 x 256 pixel 8-bit grey level representation of the fingerprint image. This data was then passed *via* a FIFO to the image processing ASIC, Parcor1, where it was smoothed and binary thresholding applied. Off-chip memory was provided to buffer the grey level data during threshold calculation and to store the final, normalised, binary image data. In principle, the same function can be implemented without the need for an ADC, FIFO, or grey-level image buffer. Figure 5.5 shows a simplified schematic of the image normalisation sub-system used in ASIP.

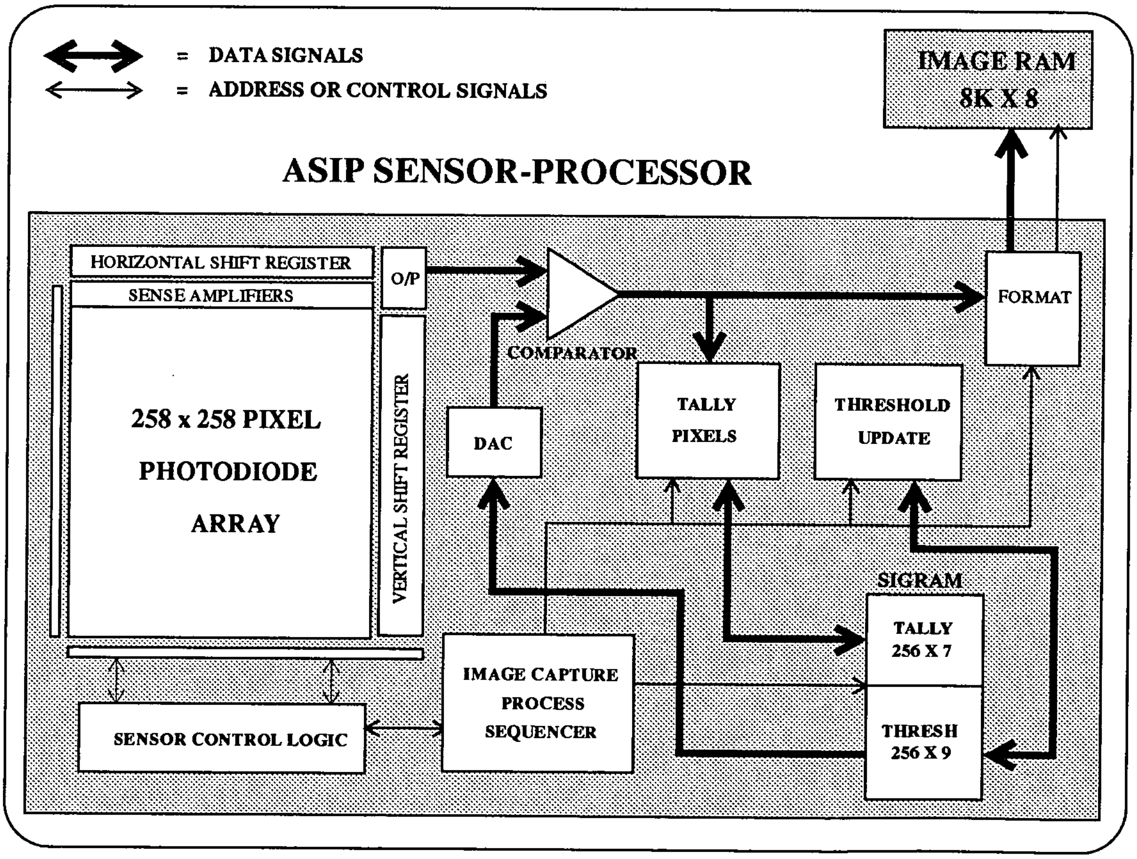


Figure 5.5 Image Sensor and Preprocessor Block Diagram

The process operates in the following manner. A sequence of nine sub-sampled image frames are read out from the CMOS sensor. This data is processed to produce 9-bit local thresholds, one for each of the 256 16 x 16 pixel patches, covering the full image space and using a successive approximation technique. Before the first frame in a sequence is accessed, each location in the threshold RAM is reset to a seed value. As the sensor array is scanned out, a threshold is driven, *via* a fast digital-to-analogue convertor (DAC), onto one input of a comparator while the analogue image signal, from the sensor, is applied to the other. The resulting binary pixel stream is analysed to determine the ratio of black to white pixels in each patch, using a tally circuit. These tally values are also buffered in on-chip RAM.

After each frame, the 256 patch thresholds are updated according to the tally results, ready for the next sub-sampled frame to be scanned out. This allows a 9-bit threshold to be calculated for each patch, 1-bit *per* frame, in nine frames. By suitable sensor addressing and sub-sampling (appropriate to this application) the calculation of the thresholds has been achieved within in one frame time.

A second, full 256 x 256 pixel frame (nine sub-sampled frames) is read and the local thresholds are applied, resulting in the required digitised binary image. Finally, the normalised binary data is formatted and stored in off-chip memory, ready for the start of the comparison process. The format for the binary image data is the same as that used in the prototype FVU.

The 3 x 3 smoothing function, implemented as part of the digital preprocessing circuitry on Parcor1, has now become an integral part of the custom CMOS sensor operating in the analogue domain. How this is achieved, as well as description of the rest of the image capture sub-system, will be presented in the next section.

5.3.2 Sensor and Filter

An entirely novel custom CMOS sensor, based on the technology described in section 5.2, has been designed for use in the image capture part of the single-chip fingerprint image processor. The sensor architecture is shown in figure 5.6. It comprises of a 258 x 258 pixel photodiode array, sense amplifiers, custom addressing and control logic, output amplification, and built-in test structures. The normal raster-scan data-access for this type of sensor has been modified to allow groups of 3 x 3 pixels to be addressed simultaneously, their values averaged and then to be read out. This method of implementing the filter function as an integral part of the sensor structure was first suggested by Wang[71].

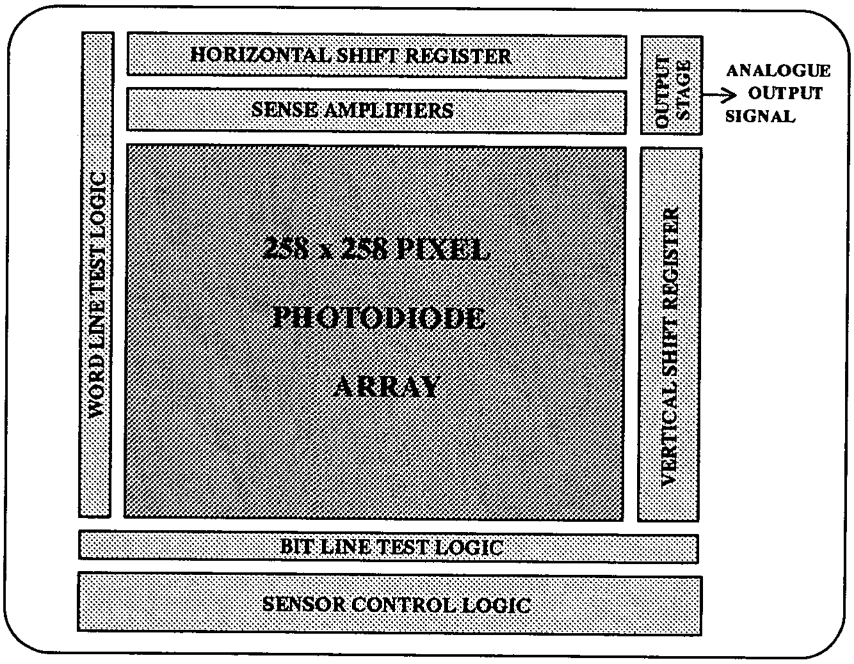


Figure 5.6 ASIP Custom CMOS Image Sensor Architecture

The averaging of a pixel with its eight nearest neighbours is achieved by selecting three word lines together, and then selecting three pixel columns simultaneously. The normal address sequence is complicated, in both the horizontal and vertical directions, by the 3 x 3 access requirement. In the horizontal direction after the first group of three pixels (0,1,2) are accessed (see figure 5.7) the next group of three would be (3,4,5) not (1,2,3) as might be expected. This is because the values of pixels 1 and 2 have already been read destructively.

A similar address pattern is also required in the vertical direction to simultaneously enable groups of three word lines. With this address pattern used in both the horizontal and vertical directions only one-ninth of the data is accessed during a single frame access. To access a full frame of averaged pixels, the array must be read nine times with a full integration period between each access. Figure 5.7 shows the pixel sample sequence during the first access, *i.e.* Frame 1.

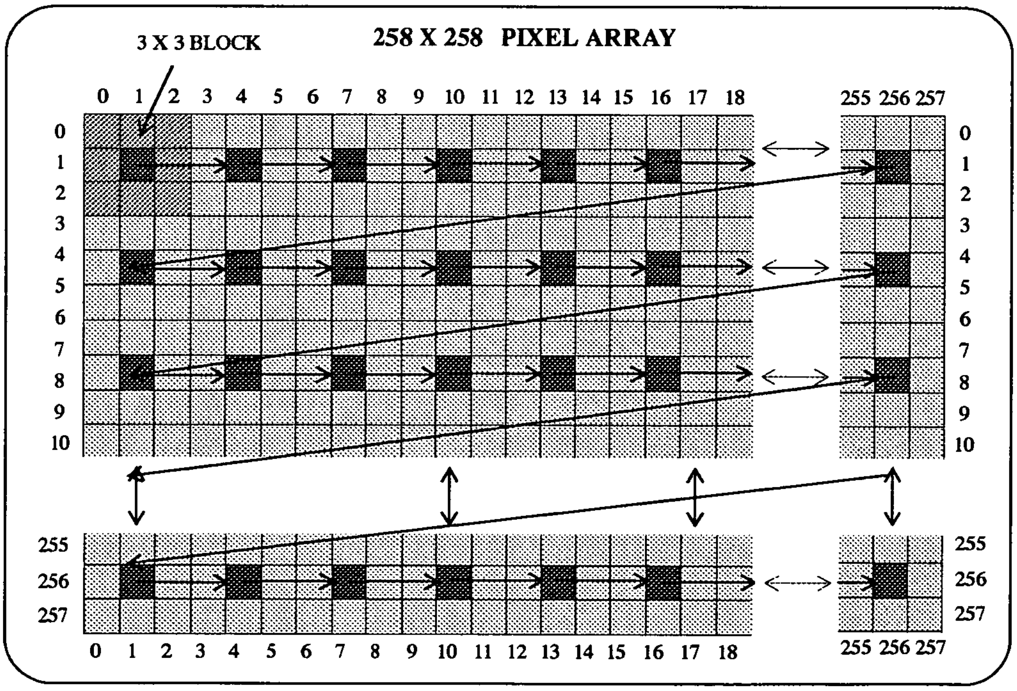


Figure 5.7 3 x 3 Pixel Scan Sequence for Frame 1

After each scan, the start position of the 3 x 3 block is moved by one pixel in either a vertical or horizontal direction and then, following the integration period, a second sequence of averaged pixel data is scanned out. After the nine sub-frames have been read out, a full locally averaged 256 x 256 pixel image has been accessed. Since the required image size required is 256 x 256 pixels, a photodiode array with 258 x 258 pixels has been chosen to eliminate spurious edge effects.

The address sequences for each of the sub-frames are as follows:

Frame 1:	Horizontal scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
	Vertical scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
Frame 2:	Horizontal scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
	Vertical scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
Frame 3:	Horizontal scan:	(2,3,4), (5,6,7), (8,9,10),...........(253,254,255)
	Vertical scan:	(0,1,2), (3,4,5), (6,7,8),.............(255,256,257)
Frame 4:	Horizontal scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
	Vertical scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
Frame 5:	Horizontal scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
	Vertical scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
Frame 6:	Horizontal scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
	Vertical scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
Frame 7:	Horizontal scan:	(0,1,2), (3,4,5), (6,7,8),............(255,256,257)
	Vertical scan:	(2,3,4), (5,6,7), (8,9,10),...........(253,254,255)
Frame 8:	Horizontal scan:	(1,2,3), (4,5,6), (7,8,9),............(254,255,256)
	Vertical scan:	(2,3,4), (5,6,7), (8,9,10),...........(253,254,255)
Frame 9:	Horizontal scan:	(2,3,4), (5,6,7), (8,9,10),...........(253,254,255)
	Vertical scan:	(2,3,4), (5,6,7), (8,9,10),...........(253,254,255)

These address sequences are generated by modified horizontal and vertical shift registers. Figure 5.8 shows the horizontal shift register logic required to produce the necessary address sequences. For normal raster-scan output there is a shift register element associated with each pixel column address bit. A line start pulse (*ls*) propagates along the shift register sequentially enabling each pixel column. In the block address scheme used here only one shift register element (*cell_0, cell_1, etc.*) is required for each group of three address lines. Control lines (*h0, h1, h2*) decode the shift register outputs to enable any consecutive group of three address lines simultaneously.

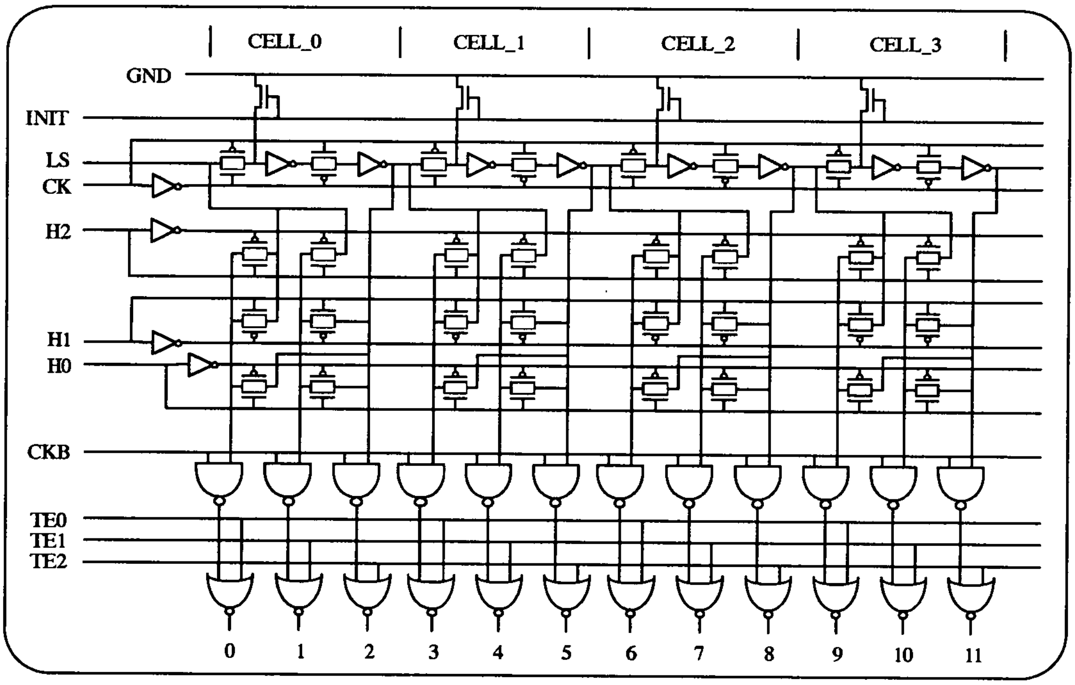


Figure 5.8 Horizontal Shift Register Logic Diagram

This allows the address patterns described earlier to be produced. A similar arrangement of is used in the vertical shift register to allow groups of three word lines to be selected together.

The field read-out time for this scheme is almost identical to the of the normal (raster-scan) method even though, to access a full image, nine sub-sampled scans are required. This is due to the fact that there is only one-third the number of delay elements in both the horizontal and vertical shift register for a given array size. Hence, the 3 x 3 unary weighted smoothing function required by the fingerprint algorithm has been implemented in the analogue domain with no changes to the photodiode array, and only a minimal increase in the complexity of the sensor control logic. This compares with the 600 gates of logic required to implement the datapath for the digital version of the filter function.

The bit-line and word-line test structures, used to test the sensor using predetermined digital patterns, have been slightly modified to accommodate the 3 x 3 block address scheme. Further control lines (*te0, te1, te2*) have been provided in the horizontal shift register to allow pixel lines to be selected individually to check for pixel column and sense amplifier faults.

5.3.3 Sensor Controller

The sensor controller produces all the clocks and control pulses necessary for the correct operation of the image sensor. The core of the controller is a divide-by-three clock generator which takes the system clock (typically 18Mhz) and produces various sensor clocks (at one-third the system frequency *i.e.* 6MHz). One of these clocks, *ck*, is used to increment a frame counter module. Outputs from this counter are decoded to generate *fs*, *ls*, sample and reset signals.

The sensor runs independently of the image capture process. Another output, *pvb*, is used by the image capture process to time the sampling of valid data from the sensor. Exposure is controlled by changing the period between reset pulses. This is known as the integration period. For this application completely automatic exposure control is not required, since the sensor is working in a controlled lighting environment. A local register allows the integration period to be set, and the exposure controlled, by the microcontroller. Another register is used to select the sensor's mode of operation *i.e.* normal or one of three test modes.

5.3.4 Sequencer

The image capture and preprocessing module, once initiated, performs the capture, filtering, binary thresholding and format functions, independently of any interaction with the microcontroller. At the core of the process control and address generator is a complex counter module, whose outputs are decoded to provide all the necessary address and control sequences. The high level control sequence is split into two phases. During *phase_1* (threshold calculation) a sequence of nine sub-frames are read out and local thresholds are calculated, one for each of the 256 16 x 16 pixel patches covering the fingerprint image space. In the second phase (binary thresholding) another nine sub-sampled frames are accessed, the local thresholds are applied to the image data and the resulting binary image is stored in the 8k x 8 off-chip SRAM.

For *phase_1* the sequencer has to generate signals to perform the following functions:

- Signal generation for the sensor shift registers, indicating which of the nine sub-sampled frames is currently being read out.
- Target threshold selection for the tally circuit. This depends on the number of sub-sampled pixels read out for a given patch in a particular frame.
- Address sequence generation for the tally and threshold RAM. This has to select the correct tally value and threshold for the next pixel to be scanned out of the sensor array.

- Timing of the update of the 256 local thresholds. The threshold is updated for a particular 16 x 16 pixel patch immediately after the last pixel of that patch has been read out and counted by the tally circuitry.

In *phase_2* the following signals are required:

- Signals for the sensor shift registers to indicate which of the nine sub-sampled frames is currently being read out.
- Format address sequences for binary image RAM. These must ensure that every binary pixel value is stored in the correct location in the format required by the comparison process.
- Signals to select the correct bit location in a byte to store the binary pixel data.

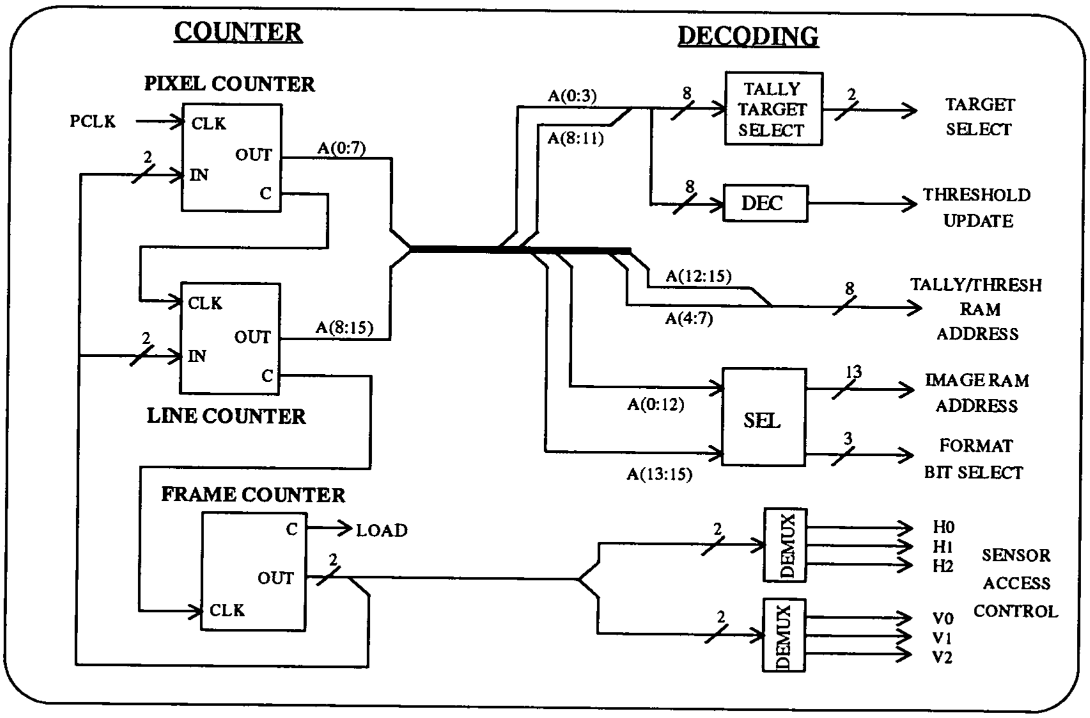


Figure 5.9 Simplified Sequencer Schematic

The sequencer consists of three counters and decoding circuitry as shown in figure 5.9. The frame counter is a modulo-9 counter, which determines which one of the nine sub-sampled frames is currently being scanned out of the sensor. At the start of a new sub-frame the pixel counter and line counter are pre-loaded with a value which is dependent on the status of the frame counter. The pixel counter is clocked each time a valid pixel is generated by the image sensor using the pixel sample clock, *pclk*. The 8-bit output of the pixel counter starts at 0,1 or 2 depending on the pre-load value and then increments by 3 after each clock cycle. A carry strobe is generated by the pixel counter when the output value is greater than 252.

This carry signal is used to clock the line counter which operates in a similar manner to the pixel counter, incrementing by three each time it is clocked. After a carry is generated, the pixel counter re-loads the frame value and starts counting again. When the line counter generates a carry strobe the frame counter is incremented and the count sequence is repeated until nine sub-frames have been processed. Higher level control logic then repeats the whole count sequence for *phase_2* of the process, binary thresholding. The image capture and binary thresholding process timing is calculated as follows:

Sensor clock cycles = frames x [sub-frames x [[delay + pixels] x lines]]
= 2 x [9 x [[42 + 86] x 86]] = 198 144 cycles

The delay before each line is accessed is required to allow video formatting to be added to the sensor output signal. In any future design this delay could be reduced to the minimum number of cycles required to select and sample a row of pixels.

The sensor clock (or pixel clock) runs at one-third of the system clock rate so the total number of system clock cycles required to capture and preprocess a fingerprint image is 198144 x 3 = 594432 cycles. With an 18MHz system clock the process execution time is approximately 33ms. Therefore, problems due to image movement during image capture do not occur, and the dynamic use of the sensor as an image storage device is vindicated.

5.3.5 Tally and Threshold Calculation

A local threshold is calculated for each of the 256 16 x 16 patches covering the image space. The 9-bit thresholds are formed 1-bit *per* frame in nine image frames.

At the beginning of *phase_1*, a start or seed threshold (50% of full scale *i.e.* 256) is set for each patch. A sub-sampled frame is then scanned out from the sensor, and the analogue output signal is compared with an analogue representation of a digital threshold, using a fast comparator. A 9-bit DAC has been custom designed for this application. Its design and implementation are described in chapter 6.

The selection of the correct local threshold, from the 256 possible for the current pixel being scanned out, is a non trivial task. After every five or six pixels a different local threshold has to be selected and converted. The datapath sequencer (section 5.3.4) performs this task. The resulting binary pixel stream is then analysed and a running total or tally of white pixels is kept for each patch.

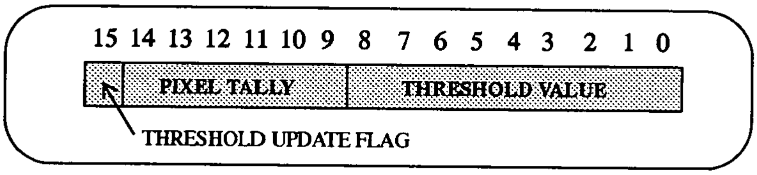


Figure 5.10 Tally and Threshold Memory Format

An on-chip 256 x 16 bit SRAM provides buffering for both the 9-bit thresholds and the tally values. Each word in the memory contains the current threshold and current pixel tally for a particular patch in the format shown in figure 5.10.

The tally scores are continuously compared with one of three target values which have been pre-loaded into local registers. These target values are based on the number of white pixels in a patch necessary to give the desired ratio of black to white pixels in the final binary image (usually 1:1). The actual number of pixels sampled in a patch depends on the start position of the 3 x 3 sample template. Analysis of the address sequence has shown that the number of sampled pixels in any given 16 x 16 patch is one of three values:- 25, 30 or 36. Hence the need for three different target values. For a ratio of 1:1 the corresponding pre-loaded target values would be 12, 15 and 18.

When the tally score for a patch exceeds the target, a flag is set in the tally/threshold memory. This threshold flag is then used to increment, or decrement, the current threshold value for a particular patch, in successively smaller binary steps after each sub-sampled frame. After nine frames the table of 256 9-bit local thresholds is complete. *Phase_2* of the process then applies these thresholds to a second sequence of nine frames to produce the required normalised binary image (see section 5.3.6).

A datapath has been designed to perform the tally and threshold calculation function, and is shown in figure 5.11. The tally circuitry keeps a count of the number of white pixels in the binary image data stream (*pixval*) for each patch. It also continuously compares each patch tally value with one of three target scores (*target_1, target_2, target_3*). The target scores are system-level parameters and are pre-loaded into local registers before the process is executed. The threshold update logic modifies the table of 256 patch thresholds according to the status of the threshold flag generated by the tally circuit, and a control strobe (*up*) from the process sequencer. A 9-bit shifter module is initialised to 256 at the start of a new sequence of nine sub-sampled image frames. After each frame the number is down shifted by 1-bit (*i.e.* 256 -> 128 -> 64 -> 32 -> 16 -> 8 -> 4 -> 2 -> 1). This number is used to set successively lower bits in the current patch thresholds gradually refining each local threshold in binary steps until, after nine frames, a 9-bit threshold has been calculated for each patch.

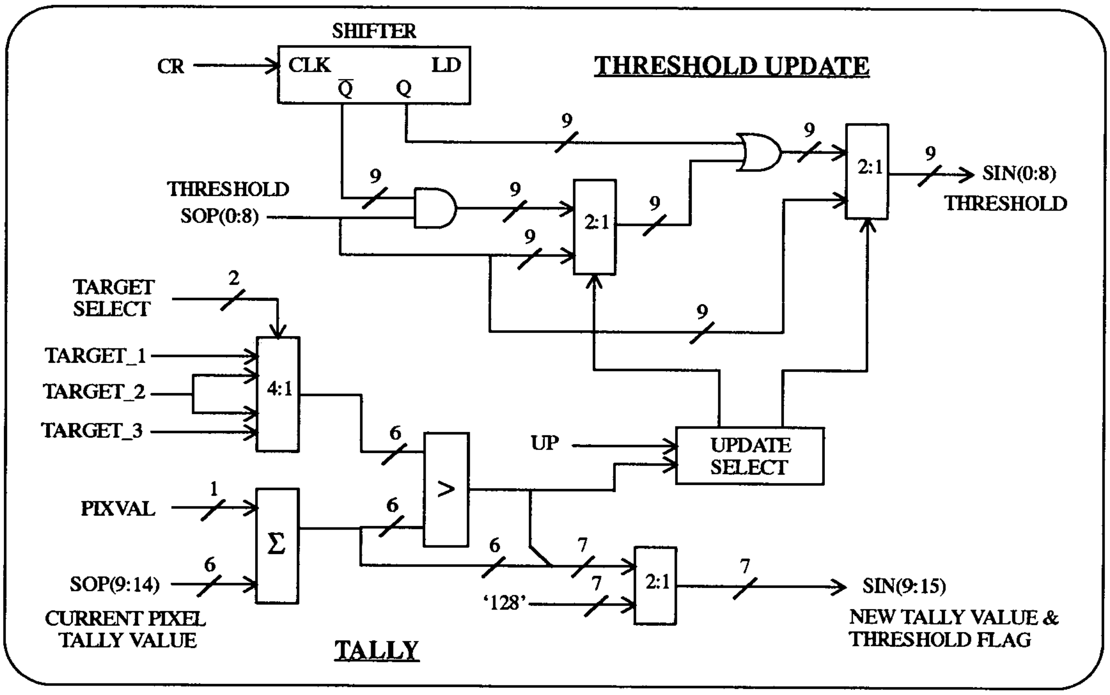


Figure 5.11 Pixel Tally and Threshold Update Datapath

This efficient mapping of the algorithm to hardware has enabled the following savings to be made, compared to the FVU system:- the total RAM requirement has been reduced from (128 x 8 + 512 x 8 + 8k x 8 + 64k x 1) = 136,192 bits to (256 x 16 + 64k x 1) = 69,632 bits and the analogue-to-digital conversion from an off-chip 9-bit flash ADC to a simple on-chip DAC and comparator.

5.3.6 Threshold and Format

After the completion of the threshold calculation phase of the process a second sequence of nine sub-sampled frames are scanned out of the sensor array. The table of threshold values are applied to the scanned data to produce a stream of binary pixel values.

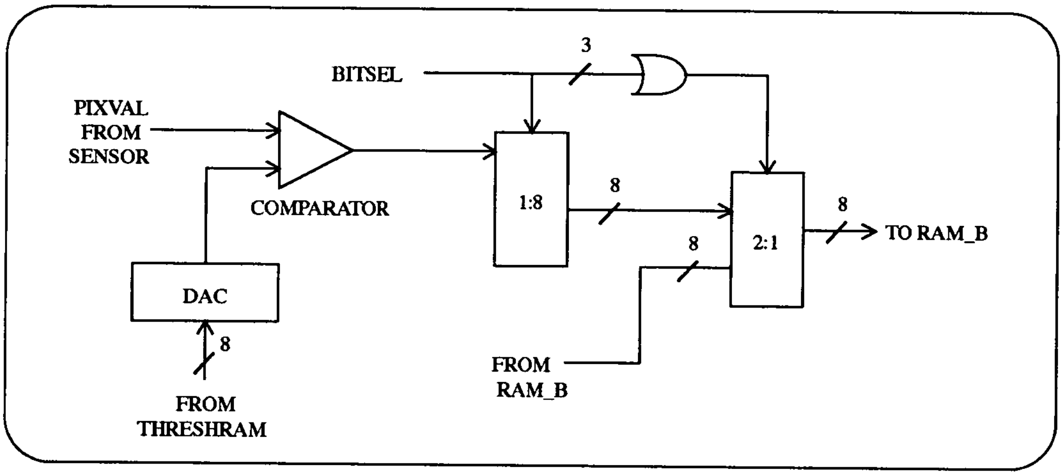


Figure 5.12 Threshold and Format Logic

During this phase, the digitised data is not counted by the tally circuit but is compressed into bytes, and stored in the format required for the next process in off-chip SRAM. The data format is the same as used in Parcor1. After each patch has been processed, its tally score and threshold values are automatically reset to their start values, ready for another image capture sequence. A simplified view of the binary threshold and data format operator is shown in figure 5.12. The required address patterns for binary image store, *RAM_B*, are generated by the process sequencer.

The image sensing and preprocessing functions are now complete and the system is ready to start the fingerprint comparison and result analysis processes.

5.4 IMAGE COMPARISON AND RESULT ANALYSIS

These processes take the captured and preprocessed fingerprint image, and compare it with a pre-stored reference using a correlation based technique.

5.4.1 Overview

The architectures used to perform the image comparison and correlation score analysis in ASIP have evolved from those used in the prototype FVU system. Changes have been made to increase throughput, and to reduce the logic gate count and the software control overhead.

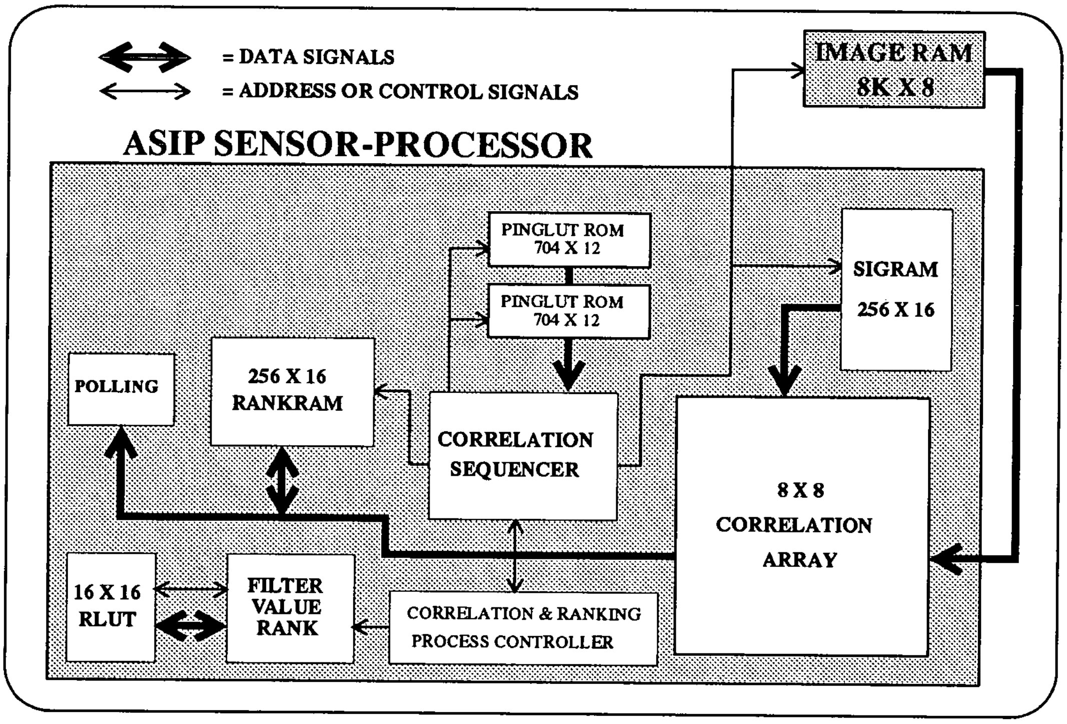


Figure 5.13 Image Comparison and Analysis Dataflow

The principle changes and additions are as follows:

- 8 x 4 correlation cell array expanded to 8 x 8 cells and integrated.
- Pipelined Correlation and Score Ranking (hardware control) has been introduced.

- Address modifier LUT ROM.
- Reference signature RAM.
- Best score table RAM.
- Pre-calculated correlation score address offset value LUT RAM.

A simplified schematic for the new architecture is given in figure 5.13. The new pipelined correlation and ranking processor has been produced by linking the individual operators by the process controller, *corrank*. On-chip memory has been provided to store reference signature data *(sigram)*, and the best score tables *(rankram)*.

The 4 x 8 correlation cell array, originally implemented as a separate ASIC (Parcor2), has now been integrated and expanded to form an array of 8 x 8 cells. This doubles the throughput of the correlation process, allowing eight signatures to be correlated simultaneously in eight different locations. The pipelined correlation and ranking process operates in the following manner.

1. Load x, y-offsets and rank thresholds into 16 x 16 *rlut* RAM. This can be done as soon as the system is powered up, since the data is pre-calculated and does not depend on the particular fingerprint being processed.
2. Load required reference fingerprint signature data into *sigram*. This happens after a user has claimed an identity, *via* either a PIN number or smart card.
3. Load number of correlation angles (typically 11): *Angles* register.
4. Load number of correlation positions (normally 64 for a 8 x 8 block position): *Corrtest* register.

5. Select signature bank: *Sigbank* register.
6. Load block address position: *Rowcol* register.
7. Start correlation/rank process sequencer (*corrank*) *via* the process control register, *control.*

The required number of 128 point correlations are then executed under the control of the correlation sequencer, *corrcon*, using the expanded parallel correlation cell array. The process cycle timing for the correlation process *corrcon* is calculated as follows:

System cycles = angles x block positions x [points + setup]
= 11 x 64 x [128 + 2]
= 91520 cycles

Following the completion of the correlation process, the 64 high score registers in the cell array are accessed and ranked in best score tables which are formed in local memory (*rankram*). The information stored in the tables is the same as in the prototype FVU system, but the format has been modified to reduce the memory requirements and speed up data access. *Corrank* makes repeated calls to the rank-value filter (RVF), once for each score in the correlation cell array. It uses the data in the *rlut* RAM and the *rowcol* register to generate the required offset block addresses, *xaddr and yaddr.* The RVF has been re-designed to reduce its size and increase data throughput. The architecture is presented in section 5.4.3. After all 64 scores have been ranked, a process-complete bit is then set in the ASIP system status register. The *corrank* process can then be repeated for a new block position or signature bank, after the appropriate registers have been updated.

When all the necessary correlation sequences have been completed, the final analysis of the best score tables is performed to complete the comparison process. This process, known as polling, scans the best-score tables looking for scores with block addresses in a 2 x 2 area of the 32 x 32 block grid covering the image space. The next sections describe these new architectures and their operation in greater detail.

5.4.2 Correlation Cell Array

The correlation cell array has been expanded from 4 x 8 to 8 x 8 cells, and has been integrated as part of the digital image processor logic on ASIP. The cell architecture is functionally identical to that used in Parcor2, but minor logical changes have been made to produce a more gate-efficient implementation. The number of cells actually implemented is 58, rather than the full 64, due to algorithmic redundancy discovered during trials of the prototype FVU system[23]. The 8 x 8 cell array architecture is shown in figure 5.14. When one of these redundant cell locations is addressed to read its best score register, a score of zero is returned.

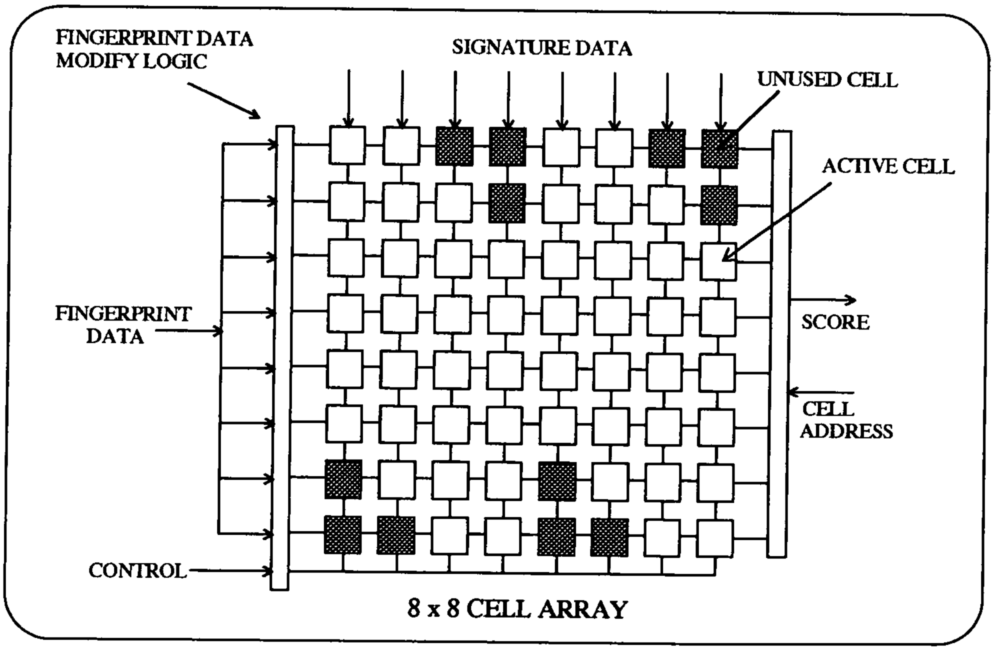


Figure 5.14 8 x 8 Correlation Cell Array

5.4.3 Rank Value Filter

The ranking process forms a table of best scores one for each of the 16 signatures. Each location in the table contains a correlation score, and a block address generated by the *corrank* process controller. The table is formed in a local memory block, *rankram*, which is realised on-chip. The new best-score ranking table format is shown in figure 5.15. The memory requirement for the best-score tables has been reduced from 8kbit to 4kbit in a 256 x 16 bit format.

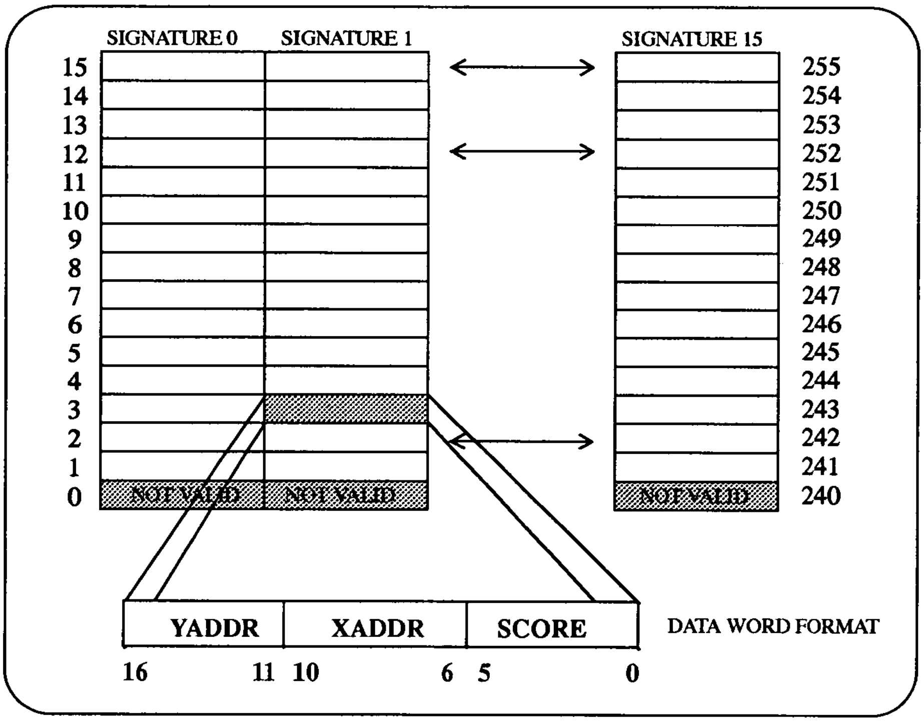


Figure 5.15 Rank Table Format

The new RVF has been designed to implement a modified ranking procedure called *dynamic ranking*. In the old RVF, a new score was compared with each score in the table, starting with the highest, until it was slotted in or the bottom of the table was reached. The modified RVF searches a table starting with the lowest value, but will terminate as soon as one of the following conditions is met:

- The new score is less than the ranking threshold.
- The new score is less than the lowest score in the ranking table.
- The new score is less than the score in the current location.
- The top of the table has been reached.

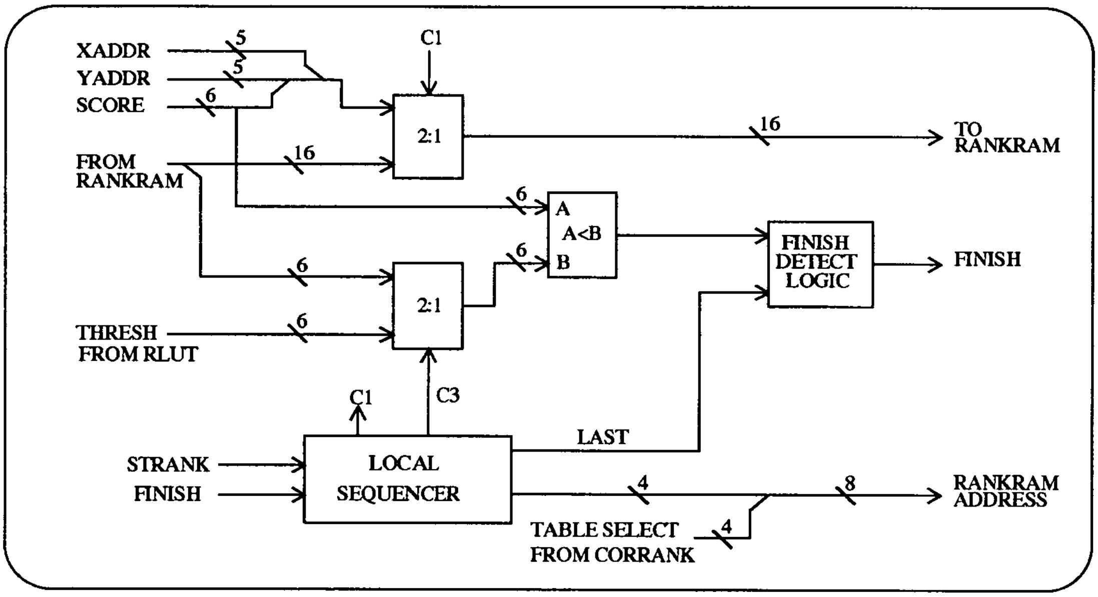


Figure 5.16 Rank Value Filter

The execution time of the ranking process is now data dependent. The simple architecture used to implement this new ranking procedure is shown in figure 5.16. Simple multiplexing of data sources onto the inputs of a comparator allows comparison of a new score with the ranking threshold and with each score read out of the best-score tables. The multiplex control signals and *rankram* addresses are generated by a local sequencer.

The cycle timing for this process is based on the number of scores to be ranked, and a variable delay dependent on which of the process-finish criteria is triggered first.

System cycles = cells x [rank_time + overhead]

= 64 x [[0 to 30] + 6]

= 384 to 2304 cycles

The 6-cycle delay between the processing of each new score is necessary to allow corrank to access a new score from the correlation cell array, and to form the appropriate offset block-address.

5.4.4 Polling

The polling process scans the best-score tables looking for scores with block addresses in a 2 x 2 area of the 32 x 32 block grid covering the image space.

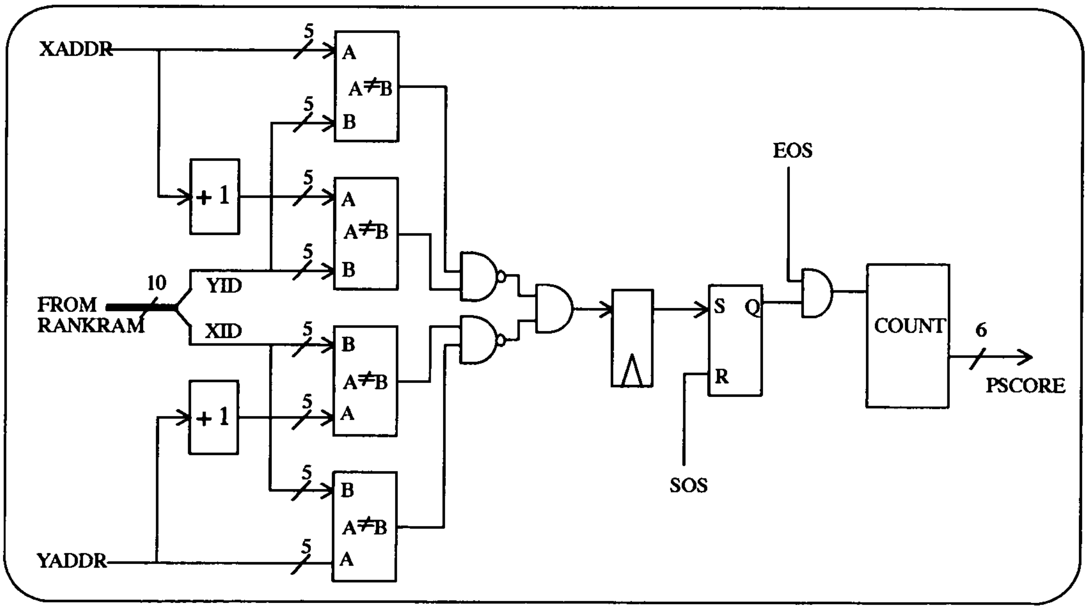


Figure 5.17 Polling Operator Datapath

The polling function architecture (see figure 5.17) is similar to that used in Parcor1, except that it takes advantage of the new best-score table format to decrease the search time by 50%. Only one best-score table memory access is now required, rather than two in the previous architecture, to read the score block address data (*xid, yid*). The start block address (*xaddr, yaddr*) has been pre-loaded into local registers. *Xid* is then compared in parallel with x*addr* and *xaddr*+1 and, similarly, *yid* is compared with *yaddr* and y*addr*+1.

The results of these comparisons are used to determine whether or not the address falls within the current 2 x 2 block defined by *xaddr* and *yaddr*. If one or more matches are found within a given best-score table, the counter is incremented by one. Hence, if at least one score exists within the current 2 x 2 search block in each best-score table, the module would return a score of 16. The polling process cycle timing depends on the number of signatures; this determines the number of best score tables, and the portion of each table which is to be searched.

$$\begin{aligned}\text{System cycles} \quad &= \text{signatures x best-scores}\\ &= 16 \text{ x } [1 \text{ to } 15]\\ &= 16 \text{ to } 240 \text{ cycles}\end{aligned}$$

High level software running, on the system microcontroller, then analyses these poll scores for all the valid block addresses. If there are sufficient scores in a given area, the system returns a positive match result. The next section will compare some of the architectural features of ASIP with the architectures used to implement the fingerprint algorithm in the prototype FVU system.

5.5 COMPARISON OF ARCHITECTURES

This section will briefly compare some of the architectural features of the prototype FVU system with those of the single chip sensor-processor ASIC.

MEMORY RESOURCES			UTILISATION			
RESOURCE	TYPE	SIZE	IMAGE CAPTURE & PRE-PROCESSING		IMAGE COMPARISON & ANALYSIS	
FIFO	FIFO	128 x 8	128 x 8	100%	-	-
RAM_A	SRAM	8192 x 8	8192 x 8	100%	-	-
RAM_B	SRAM	8192 x 8	8192 x 8	100%	8192 x 8	100%
RAM_C	SRAM	8192 x 8	512 x 8	6%	512 x 8	6%
R_LUT	EPROM	2048 x 8	-	-	1408 x 6	
C_LUT	EPROM	2048 x 8	-	-	1408 x 6	
TOTALS:		230400 bits	136192 bits	59%	86528 bits	38%

Table 5.2 FVU (PARCOR) Memory Resource and Utilisation

Memory utilisation is one useful measure of the implementation efficiency of an image processing system. Table 5.2 and table 5.3 show the memory resources used by the prototype FVU system, and the single chip ASIC. As can be seen from the data, the overall memory requirement has been reduced from 230,400 bits, in the FVU, to 90,880 bits, in an ASIP based system. This represents a memory reduction of nearly 61%. The savings have been made through architectural optimization, and the use of compiled memory cells tailored to the needs of the algorithm. The average utilisation of the available memory resources has been almost doubled from 48.5% to 88.5%.

MEMORY RESOURCES			UTILISATION			
RESOURCE	TYPE	SIZE	IMAGE CAPTURE & PRE-PROCESSING		IMAGE COMPARISON & ANALYSIS	
SIGRAM	SRAM	256 x 16	256 x 16	100%	256 x 16	100%
RANKRAM	SRAM	256 x 16	-	-	256 x 16	100%
RLUT	SRAM	16 x 16	-	-	16 x 16	100%
RAM_B	SRAM	8192 x 8	8192 x 8	100%	8192 x 8	100%
LUT	ROM	1408 x 12	-	-	1408 x 12	100%
TOTALS:		90880 bits	69632 bits	77%	90880 bits	100%

Table 5.3 ASIP Memory Resource and Utilisation

The architectural optimization has also helped reduce the number of gates required to implement the algorithm. Table 5.4 shows the gate counts for principal functional modules which are common to both the Parcor chipset and ASIP. Some of the savings were made by reducing the number of data storage elements within datapaths, and making greater use of the available on-chip memory.

MODULE	FVU (GATES)	ASIP (GATES)	SAVING
THRESH	2711	1062	61%
MICROINT	1105	1088	1%
CORRCON	774	648	16%
RANK	1018	350	66%
VOTEPOLL	681	528	22%
CONPART	-	518	
CORRANK	-	424	
CELL ARRAY	6962†	6965	0%

†This value is based on the assumption that two PARCOR2 correlation ASICs are utilised to produce the 8 x 8 cell array used in the ASIP sensor-processor.

Table 5.4 Process Gate Costs

The largest reduction in the number of gates is for the smoothing and adaptive threshold functions. The 66% saving is due to the elimination of the smoothing filter datapath, and the simplified adaptive threshold operator achieved by improved dataflow timing in ASIP.

	FVU Chipset	ASIP Sensor-processor
PROCESS	CYCLES	CYCLES
Image capture & pre-processing	209,664	594,432
Correlation	10,250,240	5,125,120
Ranking	709,632	75,264
Polling	393,216	131,072
	TOTAL = 11,562,752	TOTAL = 5,925,888

Table 5.5 Comparison of Processing Time

The time to capture and produce a binary thresholded version of a fingerprint image in the FVU system is one image-sensor frame time. In the ASIP system, the same function now takes two frame times. The first frame to calculate the threshold values from the image data, and the second to apply them to the image. This does not adversely affect the system performance since it is the correlation function, not image capture and preprocessing, that dominates the overall process timing (see table 5.5). With a 12MHz system clock the total time to capture and compare a fingerprint, excluding any software overhead, is 0.96 secs. for the FVU chipset and 0.49 secs. for ASIP. This reduction is almost entirely due to the increase in architectural parallelism, with the doubling of the number of correlation cells from 32 in Parcor2 to 64 in ASIP. Using the designed system clock rate of 18MHz, the capture and compare time for ASIP is reduced to 0.33 secs.

As well as reducing the overall memory requirements, gate count, and processing time, the new architecture dramatically reduces the system component count. Table 5.6 provides a comparison of the image capture and comparison sub-system component requirements for the FVU system and a system, using the ASIP chip. It can be seen that, by integrating the image sensor and memory, the large number of peripheral support circuitry disappears. This leads to a more compact system with lower power consumption and, potentially, greater reliability.

	FVU	ASIP
VLSI/MSI	16	3
LSI	37	0
Discretes	400 (approx.)	20

Table 5.6 System Component Requirements

5.6 SUMMARY

This chapter has presented an architecture for a single-chip sensor-processor for fingerprint comparison. The architecture has been compared with the ASIC-based prototype fingerprint comparison system for various characteristic measures including memory requirement and utilisation, gate costs, processing time, and number of components. In every respect the new architecture outperforms the existing system. The next chapter describes the method used to design and implement the ASIP architecture as a single-chip VLSI device.

Chapter 6. IMPLEMENTATION

6.1 IMPLEMENTATION METHODOLOGY

To successfully implement the architecture presented in the previous chapter it is necessary to combine two different design methodologies - full-custom and silicon compilation. The traditional full-custom approach is appropriate for creating handcrafted mixed signal (analogue and digital) circuits, but does not allow fast generation of large quantities of digital processing logic.

Silicon compilation provides an environment for the capture, simulation, layout, and verification of complex digital circuits with the design emphasis (from the engineer's point of view) on the high level design capture and simulation, rather than the low-level implementation details. The SOLO 1400 silicon compiler was used to produce the image processing ASIC, Parcor1, which was described in Chapter 4. The software package does not provide facilities for full-custom cell development or mixed-signal circuit simulation. It does, however, provide a mechanism for the integration of blocks, such as RAM, ROM, and PLA, generated by its own megacell compilers. This method of linking internally generated cells into the standard SOLO physical design flow has been adapted to allow externally created custom blocks to be included as well.

Several mixed full-custom and compiled ASIC designs have been successfully implemented at Edinburgh University. The general design flow for this method of generating ASICs is presented in figure 6.1 and has been fully documented elsewhere[96],[97].

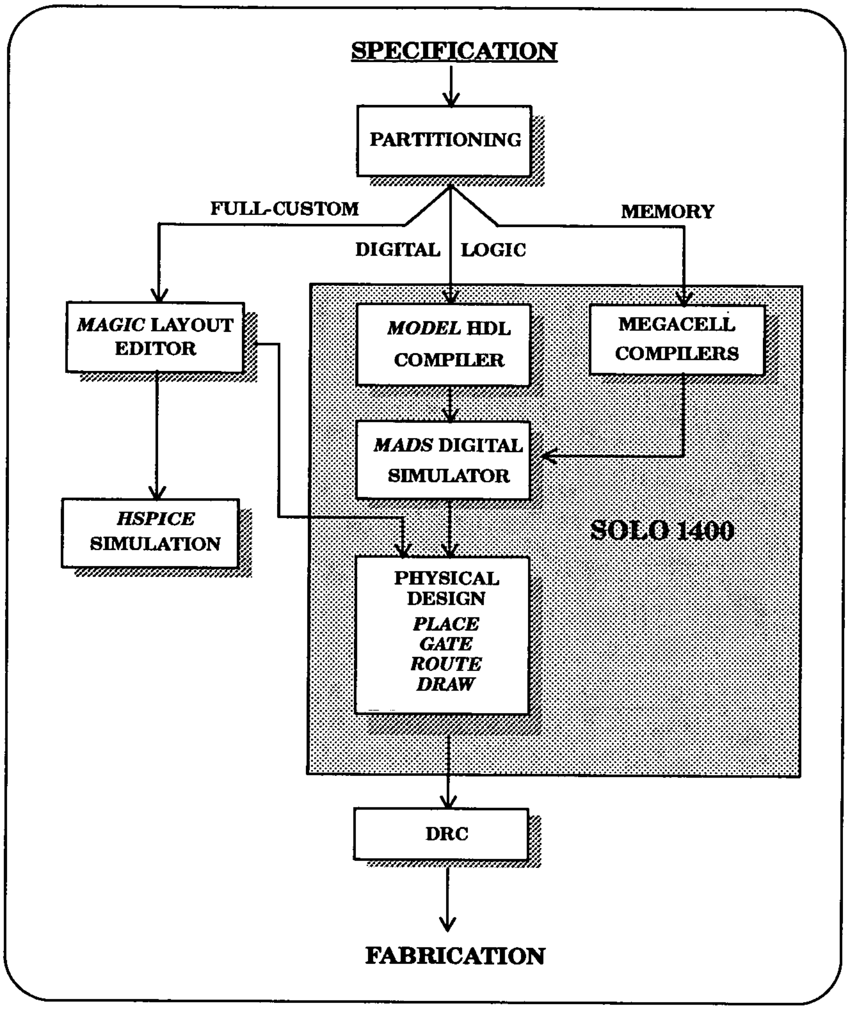


Figure 6.1 Mixed Full-Custom/Compiled Design Flow

Problems associated with this approach include:

- Different design capture techniques for full-custom and digital parts of the ASIC.
- No mixed-signal simulation.
- No simulation model of the custom block in the digital design environment.
- No post layout design verification (except basic electrical and design rule checks (ERC and DRC)).

A comparison of different methods of mixing analogue and digital ASIC design has been made by Hesketh and Burrows[57]. They suggest various improvements to this design approach, all of which are linked to the development of tools that integrate the design, simulation, and verification of full-custom cells into the compiled logic design flow. Unfortunately these tools were not available when ASIP was being designed.

6.2 DESIGN PARTITIONING

The various functional elements forming the single chip image processor-sensor can be partitioned into three groups, based on different implementation methodologies. The three methodologies are; compiled digital logic, compiled megacells (RAM, ROM, *etc.*), and full-custom cells (handcrafted digital or analogue circuitry). Table 6.1 shows the results of this partitioning exercise. As well as the full-custom cells for the sensor, DAC, and comparator, several custom input/output (I/O) pads are also required. They provide certain analogue input and output signals associated with the sensor, including isolated digital and analogue power supplies. These cells are not part of the compiler generated padring which contains the I/O pads for the rest of the chip.

The remainder of this chapter describes the implementation, simulation and verification of each of the major sub-system forming the architecture presented in Chapter 5.

COMPILED DIGITAL LOGIC

MODULE	DESCRIPTION
CONPART	. Sensor controller and image capture process sequencer.
THRESH	. Tally, threshold and format operators.
MICROINT	. Microcontroller interface.
CELL64	. Correlation cell array.
CORRANK	. Correlation and ranking process sequencer.
CORRCON	. Correlation operator.
RANK	. Rank-value filter operator.
VOTEPOLL	. Polling operator.

COMPILED MEGACELLS

BLOCK	TYPE	SIZE	FUNCTION
SIGRAM	RAM	256 X 16	Tally and threshold values or reference signature data.
RANKRAM	RAM	256 X 16	Best correlation-score tables.
RLUT	RAM	16 X 16	Block address offset modifier and ranking thresholds.
PINGLUT	ROM	704 X 12	Image rotation address modifiers for
PONGLUT	ROM	704 X 12	image comparison.

FULL-CUSTOM CELLS

BLOCK	DESCRIPTION
PHOARY	. 258 x 258 pixel photo-diode array.
HORARY	. Horizontal shift register.
SENSHARY	. Sense amplifier array.
VERARY	. Vertical shift registers.
VVV	. Bit-line and word-line test structures.
OUTPUT	. Output amplification and format logic.
DACOMP	. Video-rate 9-bit DAC and comparator.
PADBLK	. Custom I/O bonding pads.

Table 6.1 Design Implementation Partitioning

6.3 DAC AND COMPARATOR

The DAC and comparator form part of the digitisation and thresholding circuit. Threshold values are applied to one input of the comparator *via* the DAC while the analogue image data is applied to the other input. The output of the comparator is digitised binary image data. The operation of this process is fully described in section 5.3.

6.3.1 DAC Architecture

The states (on or off) of a series of nine, geometrically ratioed, n-type transistors is determined by the value of the 9-bit threshold driven onto their gates. These devices are used to sink varying amounts of current sourced from a load p-type device. As the current increases the voltage measured at the output node decreases.

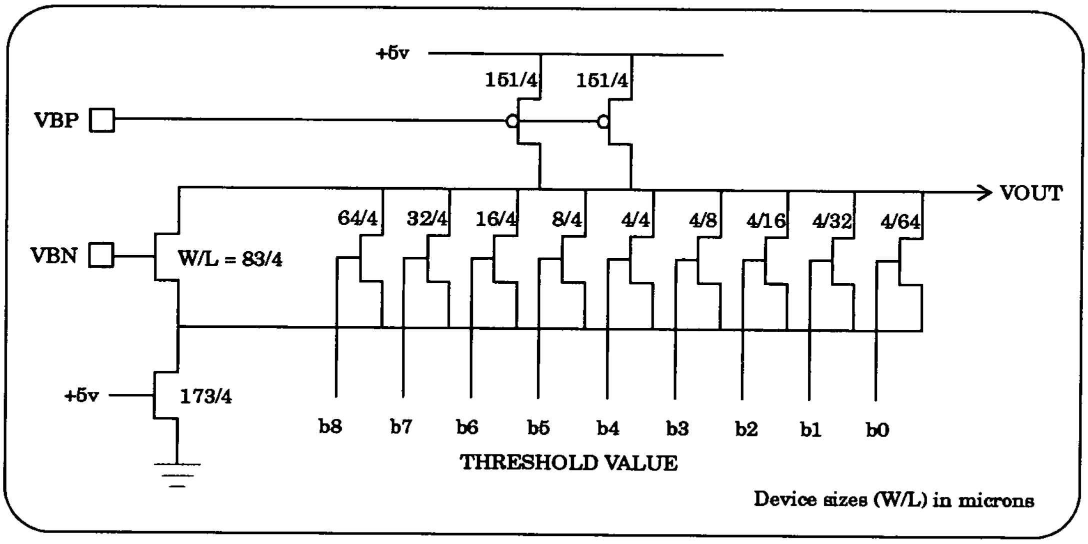


Figure 6.2 Digital to Analogue Convertor Schematic

The n-type device between the ratioed transistors and ground offsets the transfer function from zero. Further output offset and range control is provided by varying the gate voltages on the p-type load transistor and the n-type device, connected in parallel to the ratioed transistors. The required DAC output range is in the region of 2V with a 1.5V offset from ground. This is dictated by the range and offset of the sensor's output signal.

6.3.2 DAC Layout and Simulation

The size of the ratioed n-type devices is initially determined by fixing the width/length ratio for the least significant transistor, and then increasing the area in binary steps for each subsequent device. Minimum device sizes for the chosen process are channel width (W) = 2μm and channel length (L) = 1.6μm. Since the active area ratio between each device is critical to the correct operation of this circuit, it is better to use larger area devices to reduce the effects of process round-off. A minimum channel width and length is set at 4μm

INPUT BIT	TRANSISTOR DIMENSIONS	
	(A) W/L μm	(B) W/L μm
LSB b0	4/4	4/64
b1	8/4	4/32
b2	16/4	4/16
b3	32/4	4/8
b4	64/4	4/4
b5	128/4	8/4
b6	256/4	16/4
b7	512/4	32/4
MSB b8	1024/4	64/4

Table 6.2 Transistor Channel W/L Dimensions

Column A in table 6.2 shows least significant transistor set to the minimum W/L dimensions *i.e.* 4μm/4μm. Each successively more significant device doubles in area, so that the largest device has a width of 1024μm. This would give a total active area for the nine devices of 8176 μm^2. A more practical approach to implementing these devices is given in column B. The mid range device, b4, is given the minimum dimensions of 4/4. The channel width for the increasingly significant devices are modified in binary steps. Similarly, the channel length for each less significant device is modified in binary steps. Given this approach the total active area for this arrangement is 976 μm^2, *i.e.* 88% smaller. The circuit was implemented using the symbolic full-custom design tool, Magic[102],[102], with the device sizes as given in figure 6.2.

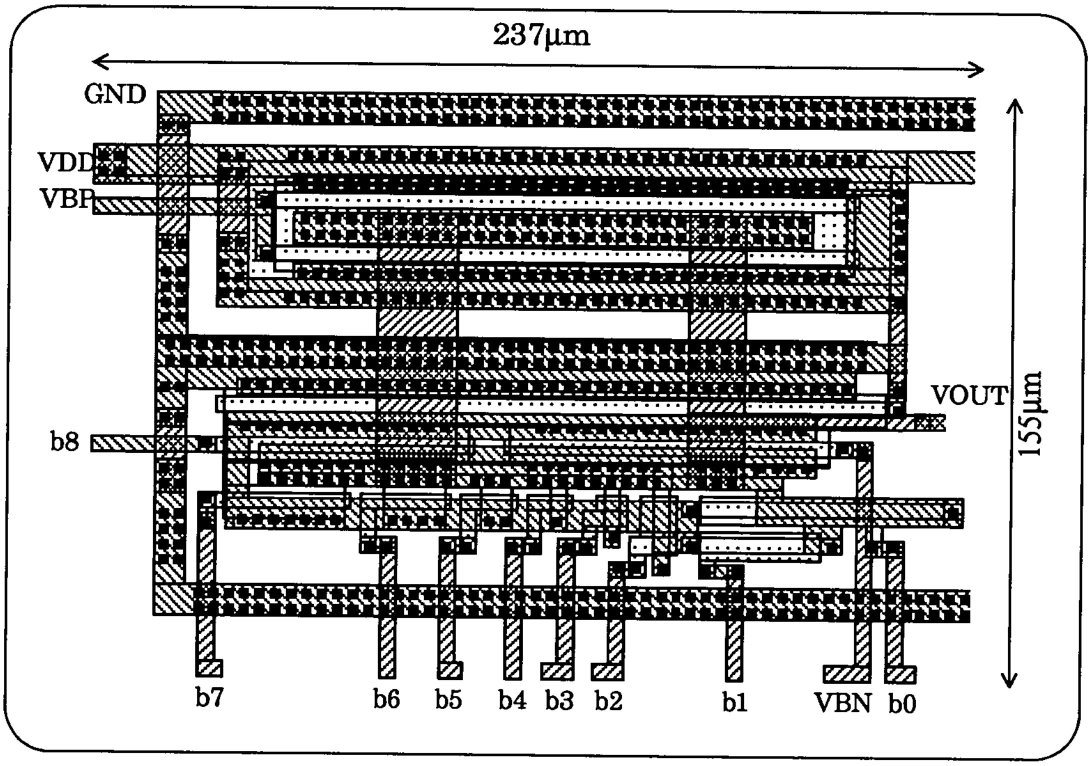


Figure 6.3 DAC Layout

The final circuit implementation is shown in figure 6.3. A guard ring connected to VDD has been placed around the p-type devices, while a second ring connected to GND has been formed round the n-type devices.

A simulation model is extracted from this circuit and its operation verified using the HSPICE simulator[100] with Level 6 process models. Starting with the input word (b8-b0) set to zero, then cycling through all possible values, the transfer characteristic for the circuit shown in figure 6.4 is produced.

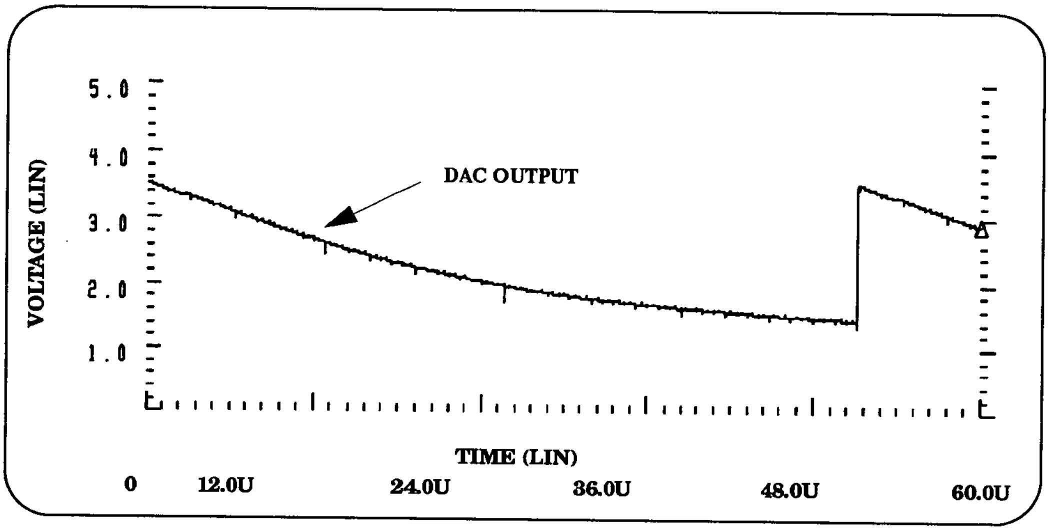


Figure 6.4 DAC Transfer Characteristic (HSPICE)

6.3.3 Comparator Architecture

The comparator compares the output from the sensor and the DAC, and produces a stream of digitised binary pixels. The comparator architecture chosen for this process is based on a standard, two-stage comparator circuit described by Allen and Holberg[93].

The first stage of the circuit is a differential amplifier, see figure 6.5, whose objective is to amplify the difference between the two input potentials. A high gain inverting stage, and simple inverting output buffer forms the second stage of the comparator. Assuming common mode input potentials, the current flowing in M5 is split equally between the paths M1/M3 and M2/M4. This current is mirrored to the second stage by the ratio of M7 to M5. The current in path M2/M4 is mirrored to M6 by the ratio of the device sizes of M4 and M6. The various devices are sized to meet the requirements of the application, which include an output slew rate of 500V/µs, an input voltage range of 1.5-4.5V, and a minimum input difference of 4mV with a 5V supply voltage.

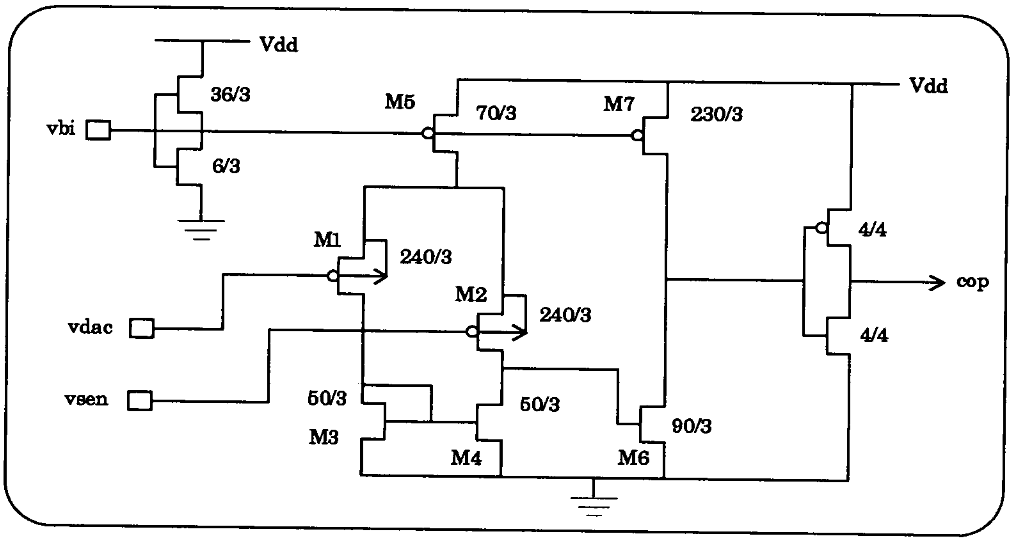


Figure 6.5 Two-Stage Comparator and Buffer

6.3.4 Comparator Layout and Simulation

Design equations given by Allen and Holberg are used to calculate the initial device sizes. Using these values, the circuit is implemented using the Magic design software, and the circuit parameters extracted for simulation. After simulation, the device sizes are modified iteratively, extracted, and re-simulated to more closely match the desired circuit specification. The final layout circuit, with its guard rings, is shown in figure 6.6.

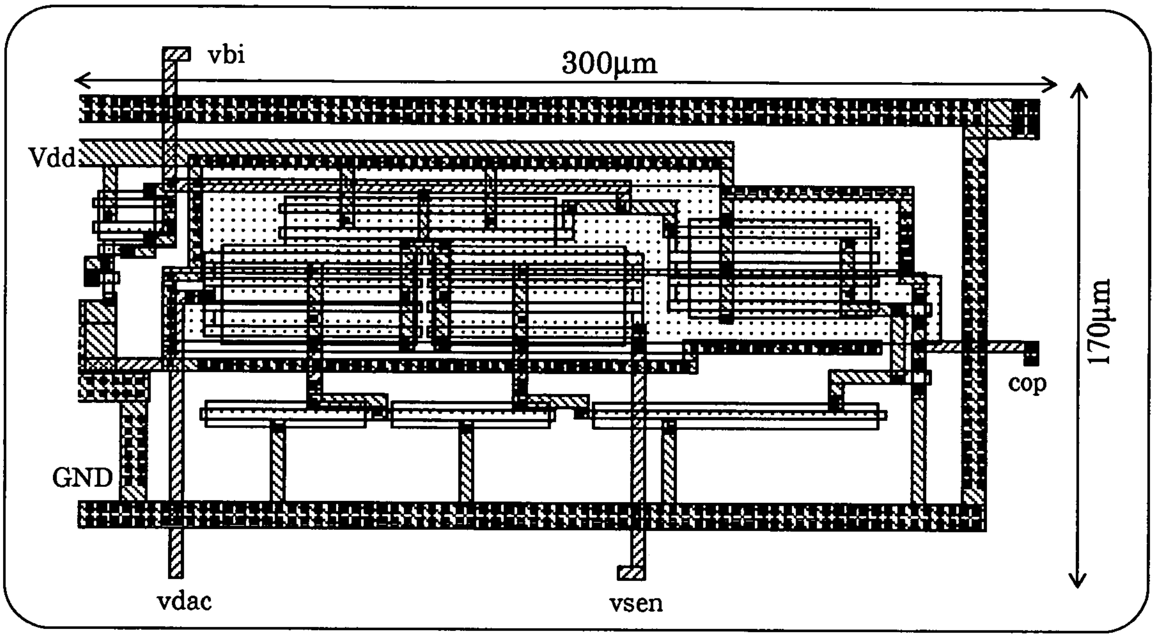


Figure 6.6 Comparator Layout

The simulation results presented in figure 6.7 trace the output of the comparator for a variety of input voltage offsets and differentials. Initially *input_a* is set to 3.49V. The second input on the comparator is set 3.48V (*i.e.* 10mV below *input_a*). The comparator output is 5V with these input levels. At 105ns, *input_b* is changed from 3.48V to 3.52V. Within 20ns of the input changing, the output has moved full-scale from 5V to 0V. At 255ns, *input_b* reverts to 3.48V and after approximately 60ns the comparator output swings back to 5V.

Input_a is then set to 1.49V and *input_b* is switched between 1.48V and 1.50V. The comparator output again switches between 5V and 0V. This simulation demonstrates the comparator's function at the extremes of its input range, with an input differential of 20mV.

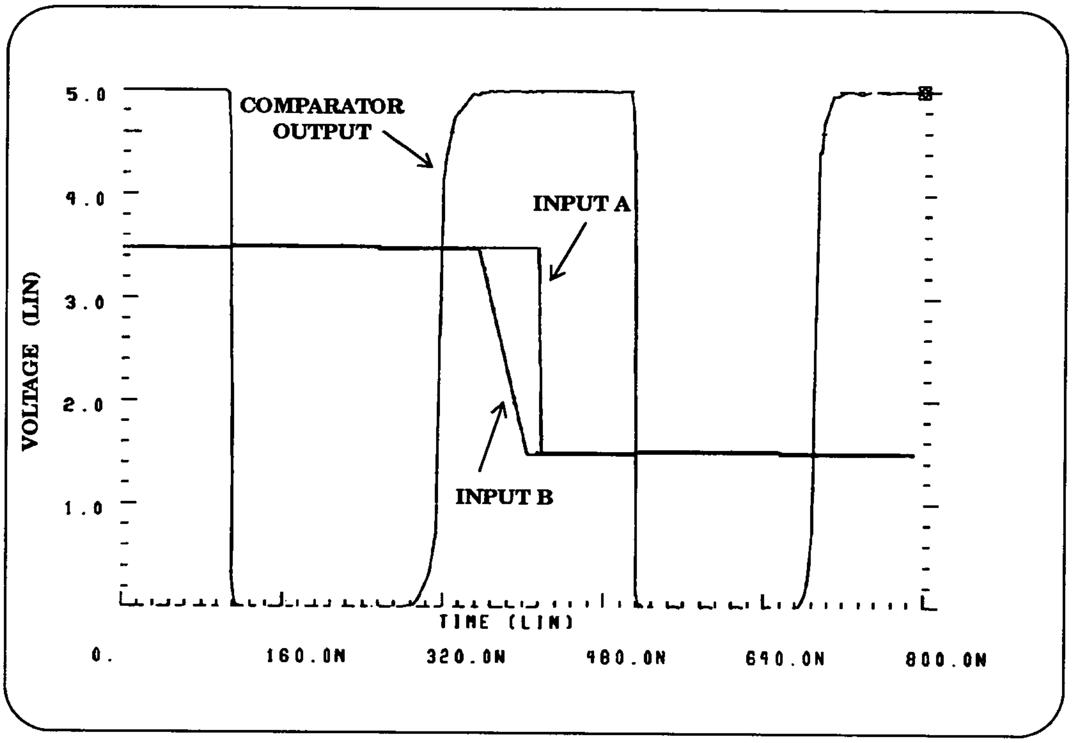
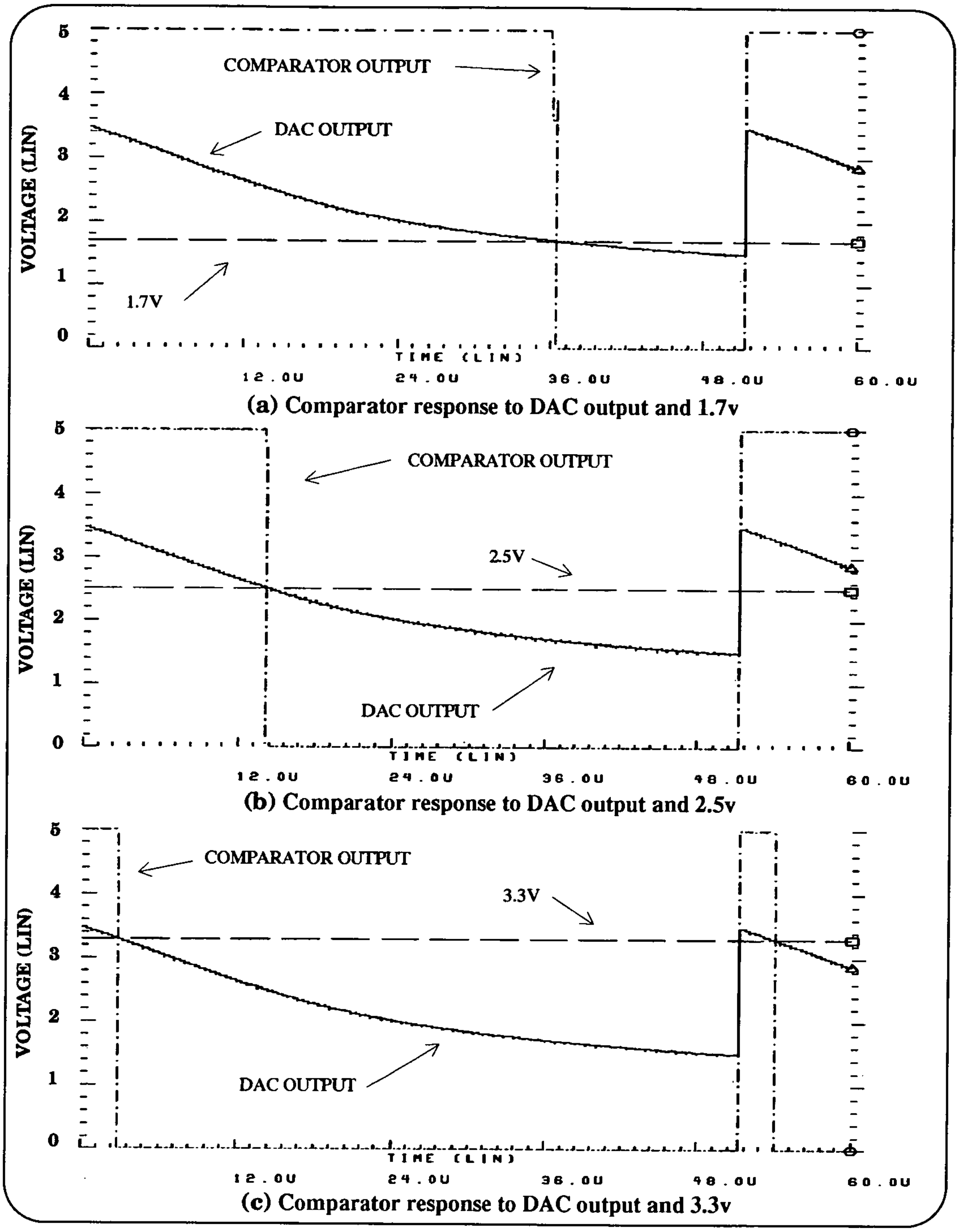


Figure 6.7 Comparator Operation (HSPICE)

The simulation results shown in figure 6.8 (a), (b), and (c) show the output response of the DAC to the full range of possible digital input values (*i.e.* 0 - 511). The graphs also plot the response of the comparator, with a fixed voltage on one input, when the output of the DAC is applied to the other input. The fixed input voltage levels are: 1.7V in (a), 2.5V in (b) and 3.3V in (c). As the DAC output matches the d.c. voltage the comparator output switches from 5V to 0V.



(a) Comparator response to DAC output and 1.7v

(b) Comparator response to DAC output and 2.5v

(c) Comparator response to DAC output and 3.3v

Figure 6.8 DAC and Comparator Operation (HSPICE)

6.4 CUSTOM SENSOR

The image sensor module consists of the following parts: 258 x 258 photo-diode array, sense amplifiers, horizontal shift register, vertical shift register, output amplifier logic, test structures. These elements, plus the DAC-Comparator block and several custom I/O pads, are formed into a single, large, custom block which is integrated with the compiled digital processing logic.

The starting point for the development of the custom sensor is a module designed as a single-chip CMOS monochrome video camera[86]. A comparison of the specification of the monochrome camera and the requirements for the custom sensor is presented in table 6.3.

FEATURE	CCIR FORMAT CMOS SENSOR	CUSTOM CMOS SENSOR
Photodiode array	312 x 287	258 x 258
Read-out sequence	raster scan	3 x 3 block scan
Pixel size	20μm x 16μm	20μm x 16μm
Output format	CCIR video	Unformatted
Exposure control	Electronic	Electronic
Clock	6MHz	6MHz
Test	Built-in test	Built-in test

Table 6.3 CMOS Sensor Characteristics

To create a image sensor suitable for this application, several parts of the existing design have to be modified. The pixel aspect ratio required by the fingerprint verification system optics is the same as that used in the monochrome sensor, *i.e.* 4:3. This means that the photodiode array can be utilised with only the array size needing to be changed, from 312 x 287 to 258 x 258 pixels. The unique 3 x 3 block read-out scheme means that the horizontal and vertical shift registers from the original sensor are not suitable and, therefore, have to be completely redesigned for this application.

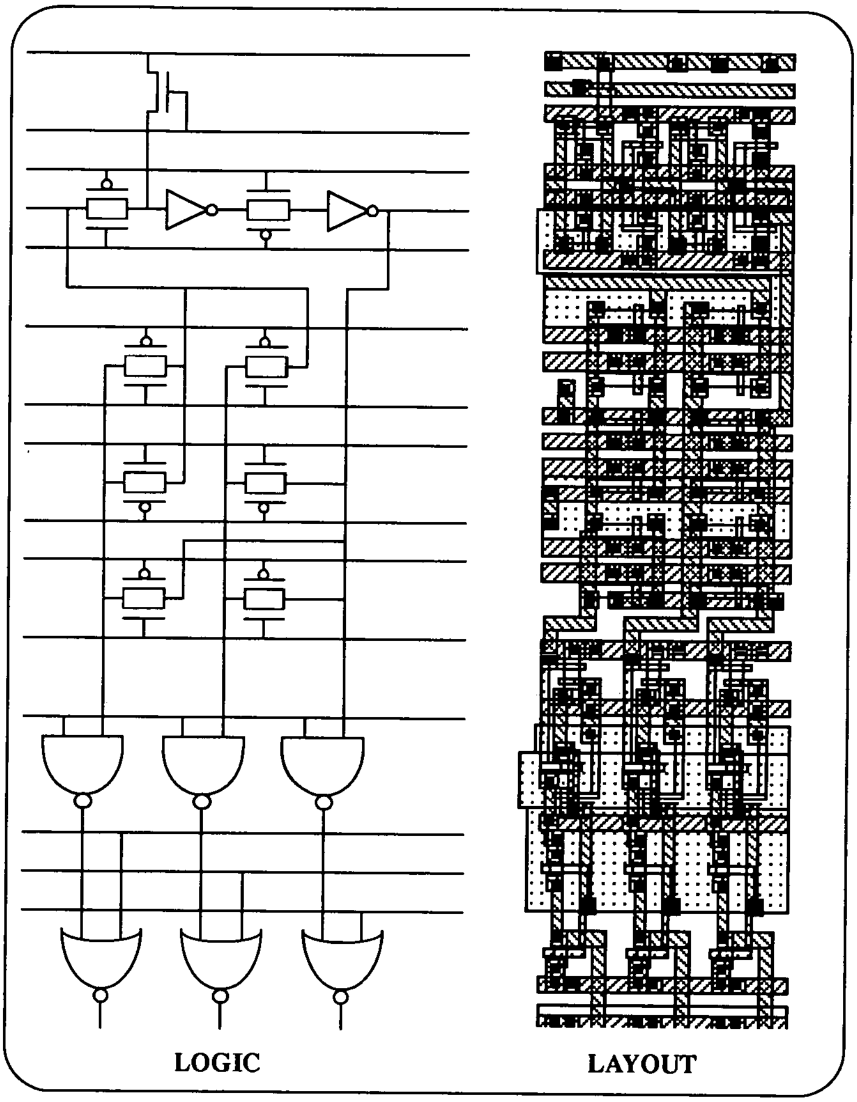


Figure 6.9 Horizontal Shift Register Cell

During normal sensor operation, the horizontal shift register is required to enable contiguous groups of three sense amplifier, output storage, capacitors to scan out a line of pixel data. The horizontal shift register circuit, logic and functional operation has been described in detail in section 5.3.2. Figure 6.9 shows the logic for a single shift register element (3 pixel columns) and its corresponding layout. Note that the circuit is pitch-matched to the width of a group of three pixels. The width of the full shift register module (*i.e.* 258 x 20μm = 5160μm) is the same for either the standard raster-scan addressing format, or the new 3 x 3 block scheme. However, the depth of each shift register cell for the 3 x 3 scheme is reduced, since only a single flip-flop is required for each group of three pixels.

During the sensor test procedure it is desirable to be able to individually address pixel columns. Three additional control lines are added to the decoding logic within a cell to enable this to happen. To verify the operation of the horizontal shift register logic, a small number of cells are linked together to form a short shift register, which is then simulated using HSPICE.

The vertical shift register controls the selection of particular lines of pixels, which are then accessed in parallel. For the 3 x 3 addressing scheme, three lines have to be selected simultaneously. Further logic is provided to control the timing of the resetting of pixels and sense amplifiers.

Another consequence of this addressing scheme is the need to modify the bit-line and word-line test structures, to maintain the same level of error discrimination as is available in the normal raster scan test procedure.

The output stage includes circuitry to perform the following functions; signal amplification, gain control, offset control, and video formatting. The video formatting circuit is not required by the fingerprint application. Since the formatting function can be disabled *via* a control signal, it is not necessary to remove or modify this module.

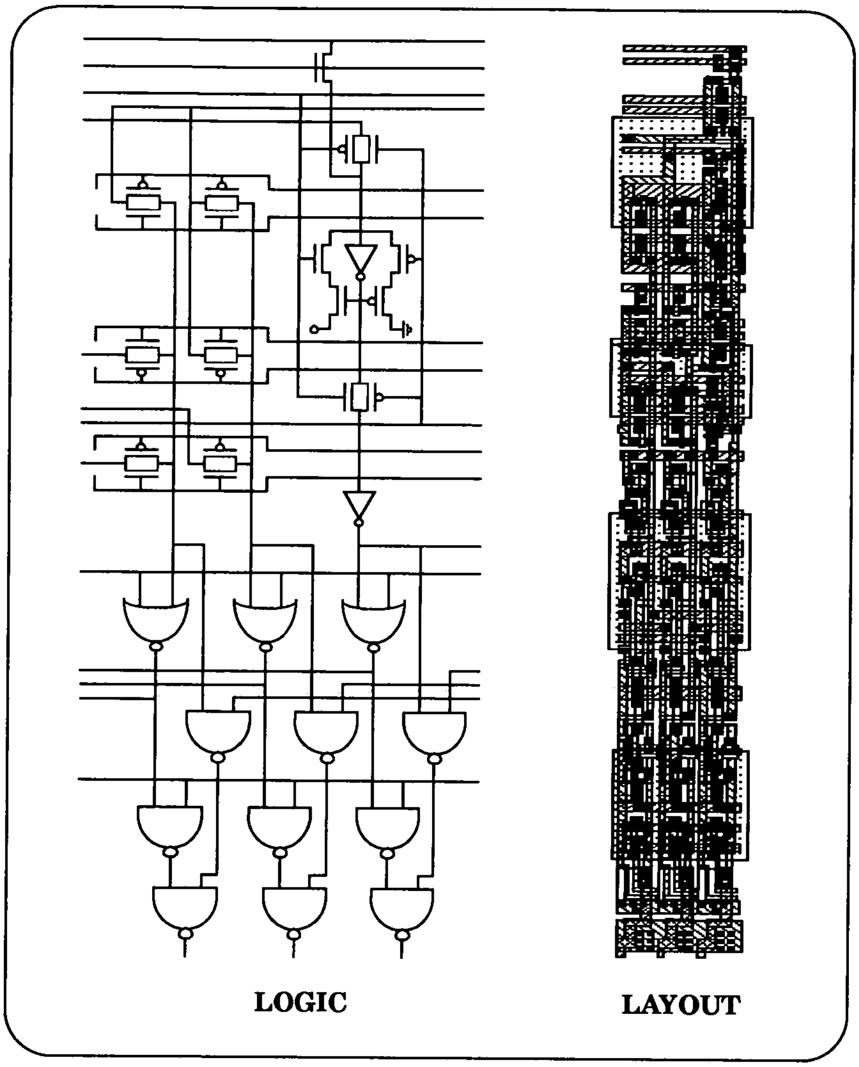


Figure 6.10 Vertical Shift Register Cell

6.5 CUSTOM BLOCK AND I/O PADS

Before the individual full-custom cells forming the sensor, DAC, and comparator circuits can be integrated into the final design, they are assembled into a single large block. Some custom I/O pads, including isolated power supplies for the sensor and

DAC, are also included in this block. They are positioned so that they will lie just within the SOLO generated pad-ring when the block is included in the chip floorplan. Figure 6.11 shows the final artwork for the custom block, highlighting the main elements.

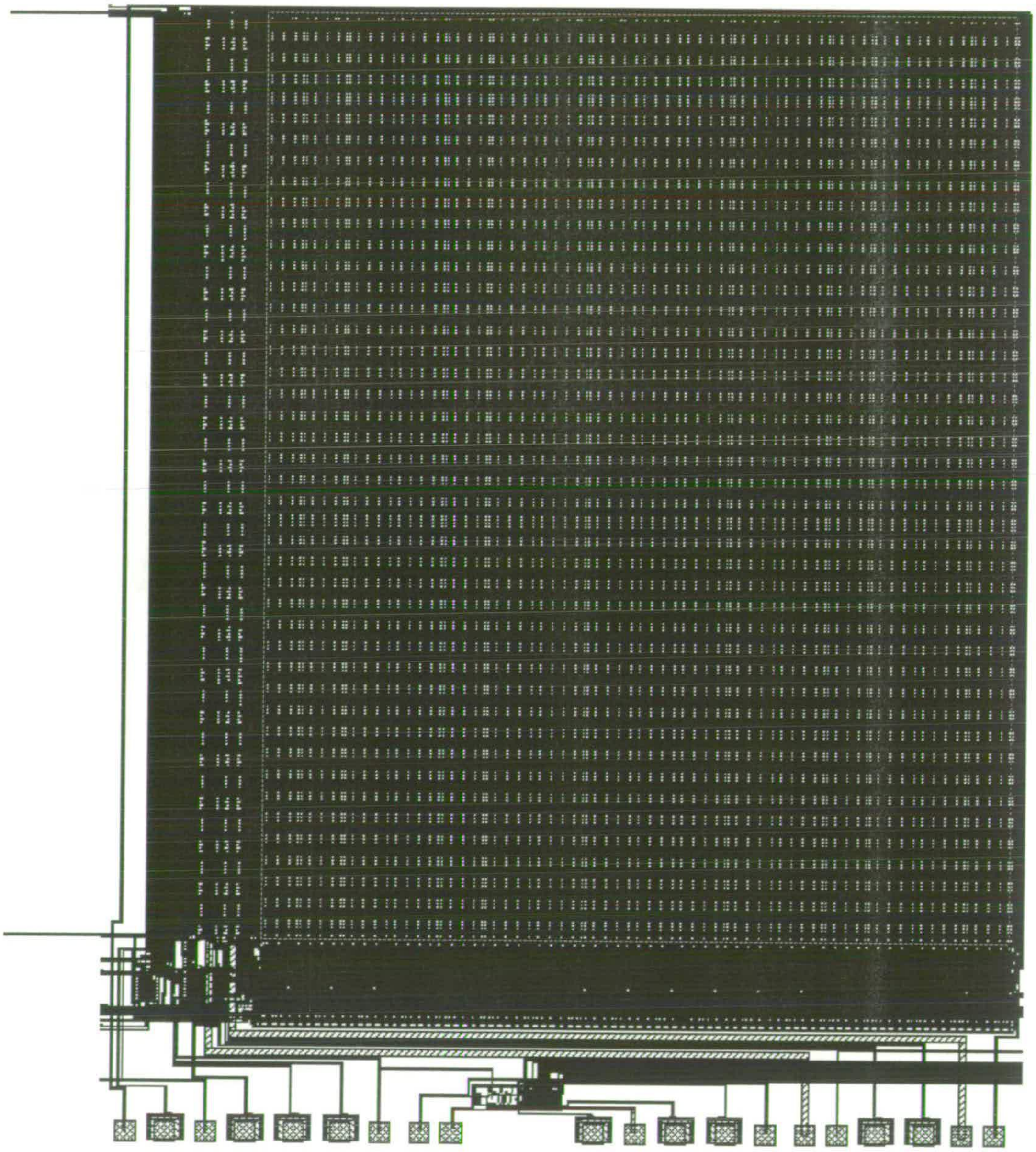


Figure 6.11 Custom Block Artwork

6.6 DIGITAL LOGIC AND COMPILED MEMORY

The implementation of the digital logic follows the standard SOLO design flow, which is described earlier (see section 4.5.1). The digital elements of the design are captured using SOLO's hardware description language, *model*. To reduce the computational load, the design is created and simulated as a series of separate functional modules. These modules are then linked together, with the necessary compiled memory and the full-custom block to form the complete ASIP chip.

The required static RAM modules are compiled using the SOLO megacell generation software. Table 6.4 provide a summary of the main characteristics of each of the memory blocks.

PARAMETER	SIGRAM OR RANKRAM	RLUT	PINGLUT OR PONGLUT
Generator:	RAM v1.2.8	RAM v1.2.8	RAM v1.2.8
Words:	256	256	256
Bits *per* word:	16	16	16
Memory enable:	inverted	inverted	inverted
Write enable:	inverted	inverted	inverted
Output buffer:	normal	normal	normal
Technology:	ECPD15	ECPD15	ECPD15
Dimensions:	3082 x 1678 mm	1610 x 554 mm	4427 x 864 mm
Area:	5.17 mm^2	0.89 mm^2	3.82 mm^2
Rotation:	90°	none	none
Cycle time (typ.)	17.61 nS	11.70 nS	40.36 nS
Power:	4.39 mW/MHz	1.96 mW/MHz	5.30 mW/MHz
BIST:	yes	yes	yes

Table 6.4 Compiled Memory Data Sheets

The worst-case access times of the ROM generated by SOLO, is not sufficient to meet the requirements of this application. The solution is to split the LUT data between two modules, PINGLUT and PONGLUT, and to read data out from each on alternate clock cycles. This effectively reduces the worst-case cycle time for the ROM from 79ns to 55ns (based on an 18MHz system clock).

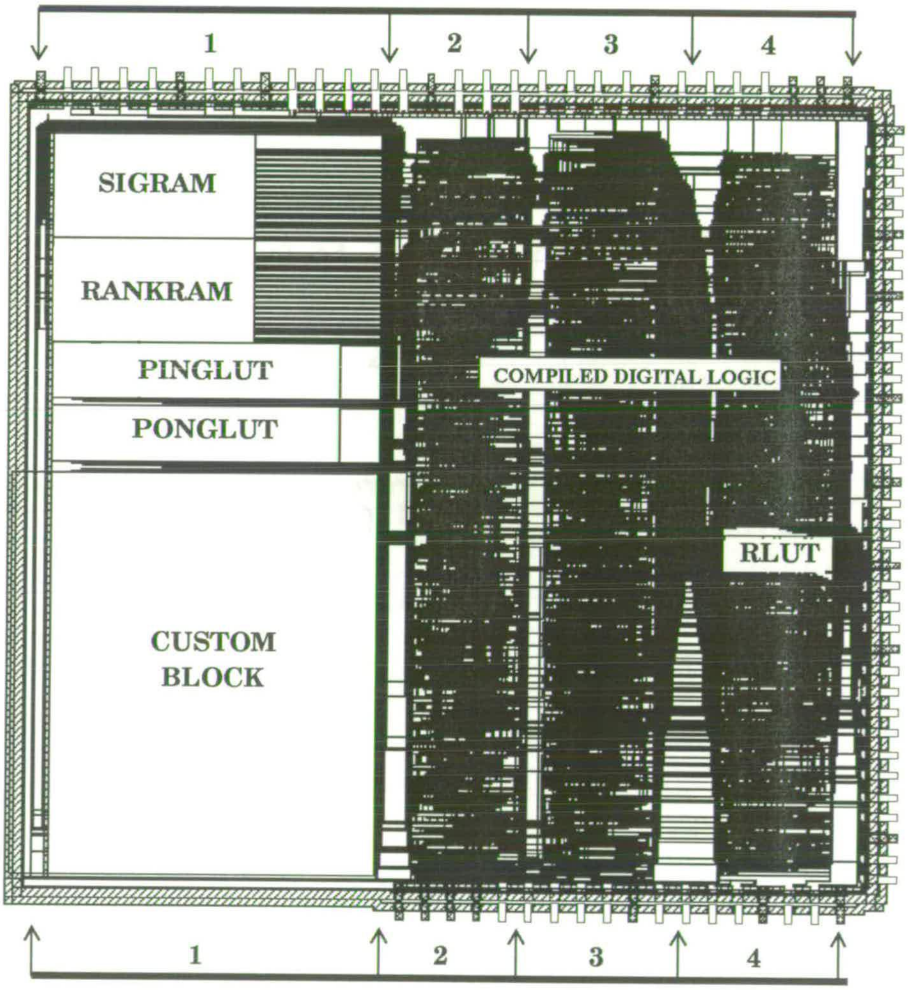


Figure 6.12 ASIP Floorplan

The system clock from the input pad and buffer is distributed to the digital logic *via* a number of high drive buffers. To help prevent clock skew problems between the various on-chip processes, the load on each clock buffer is balanced where possible.

After several iterations, the final floorplan for the design is produced as shown in figure 6.12. A slightly more compact design can be achieved by distributing the memory blocks in columns 2 - 4, and placing some of the digital logic in column 1 to completely fill the area above the custom block. Unfortunately, the width of this column would violate the row power-supply limit, imposed by the SOLO software. The lack of sophisticated floorplanning software in SOLO is a severe limitation to the efficient implementation of the proposed architecture.

The digital padring for the chip is generated interactively with SOLO. Care is needed to ensure that the area of the padring close to the custom pads is kept free of digital pads. The overall size of the chip is 14mm x 14mm giving a die area of just under 200mm^2. After completing visual checks on custom block to compiled logic connectivity, the design artwork is taped-out and shipped to ES2 for fabrication of prototype devices.

This chapter has described the mapping of the fingerprint comparison architecture to a highly integrated single-chip sensor-processor. Two different implementation methods, full-custom and compiled cell, have been combined in a single device. The simulation results for the full-custom elements are also given. The fabrication, packaging and testing of the prototype ASIP sensor-processor is described in the next chapter.

Chapter 7. FABRICATION AND TESTING

7.1 FABRICATION

Prefabrication processing of the design was carried out by ES2, during which the artwork for the compiled memory megacells and the I/O pads was generated to complete the design. A simple, metal layers only, design rule check (DRC) was executed to verify the placement of these blocks in the final design. To reduce the cost of prototyping the ASIP chip, no post-fabrication testing was carried out by the silicon vendor.

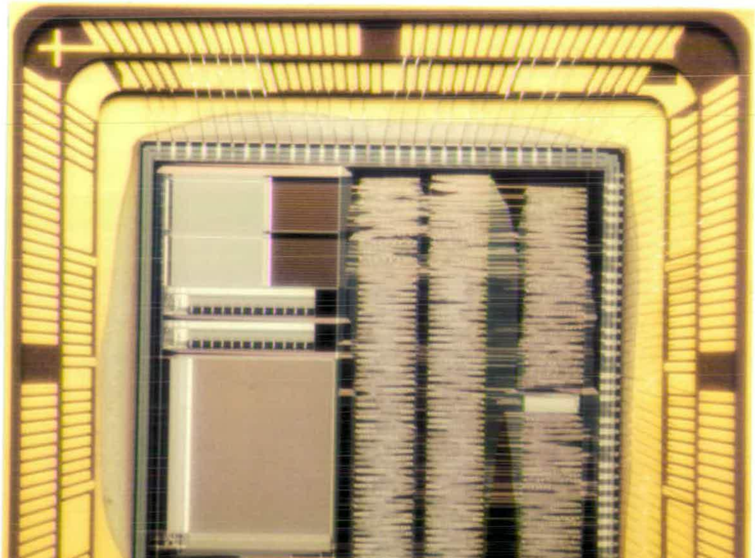


Figure 7.1 Photograph of ASIP

Due to the large size of this device, approximately 200 mm^2, relatively few packaging options were available. A custom package of some type, such as direct bonding onto a ceramic substrate, was considered but rejected, on the grounds of cost and added risk. ES2 were eventually able to supply a 299-pin, ceramic PGA, package with a well size which could accommodate the single-chip sensor-processor. The excessive number of pins (the chip only requires 105), and a downward facing well caused some problems for the design and construction of a suitable test rig. A total of twenty prototype devices were packaged. Table 7.1 provides a summary of the main physical characteristics of the ASIP sensor-processor chip.

ASIP	
Foundry	ES2
Process	1.5μm
Transistors	272 000
Dimensions	14mm x 14mm
Area	196 mm^2.
Clock (max.)	18MHz
No. I/O	105
Package	299 CPGA

Table 7.1 ASIP Physical Characteristics

7.2 TEST PROCEDURE

The equipment used to test ASIP after fabrication and packaging is shown in figure 7.2. The device under test is plugged into a simple test board, which consists of an 8051 microcontroller, image memory and the necessary power and bias circuitry. The chip is controlled *via* a microcontroller emulator, a Hitex Teletest 51[98], which provides a hardware and software environment enabling in-circuit emulation of the 8051 microcontroller. Interactive software supplied with the emulator, running on an IBM compatible PC, allows communication with the emulator and target system.

The test routines were written in ‘C’ and then cross-compiled for the 8051 microcontroller using the Keil C-51[99] compiler. The compiler additionally optimises the code for compactness and speed. The resulting code is then downloaded *via* a serial link to the emulator where it can be executed and the results monitored.

The responses of the chip to the stimuli generated by the microcontroller, were monitored using a variety of equipment. Analogue characteristics were measured using an oscilloscope while digital signals, such as address and data buses, were traced using a logic analyser.

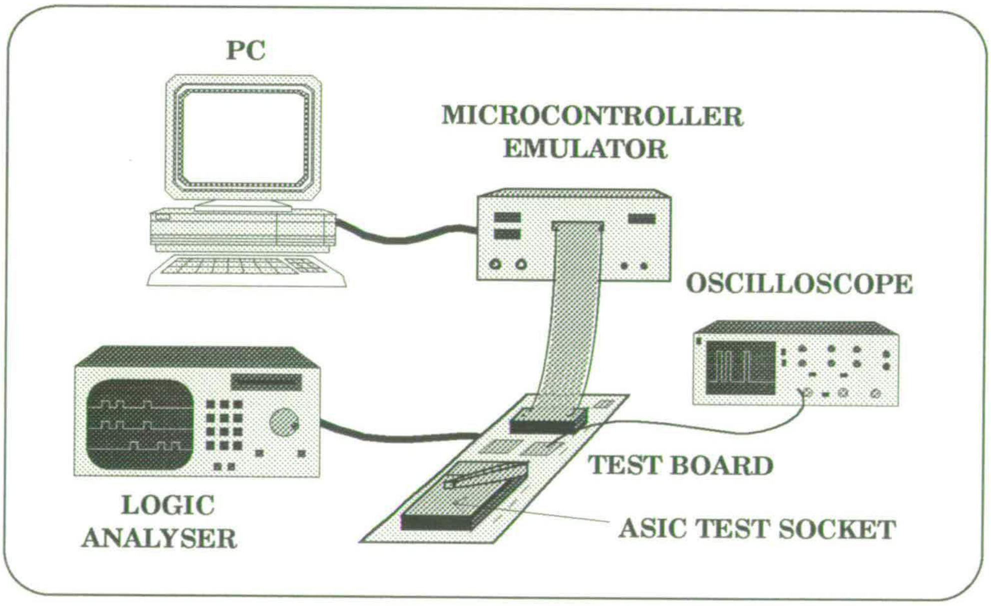


Figure 7.2 ASIP Test Setup

The testing has not been completed, but the data obtained so far is presented in the rest of this section. In total eight devices have been tested. Initially, testing was performed on the chips to check the power supply and clock generation circuits.

When the first device was powered up, and the initialisation pulses were issued, the test circuit (including ASIP) consumed around 400mA at 5V with 12MHz system clock. After a short period (around 60 seconds), the supply current fell to around 200mA. The power supply was cycled, but the device continued to draw 200mA. This suggested the existence of a low resistance path between the power rails which had failed. A second device was tested, producing a similar result. At this stage, a thorough review of the test rig, the package pinout, and device bonding was carried out but revealed no inconsistencies.

Next, the interface between the compiled logic and the custom block and the custom block itself was checked for possible power supply connection errors. This was done using the Magic layout editor. By selecting individual I/O ports, it was possible to check if tracks were electrically connected. Eventually, a short between power and ground was discovered in the output stage of the image-sensor block. This had occurred where various modules in the output circuitry of the sensor block tapped off power and ground from parallel supply rails. The ground connection for a particular module, which was connected to more than one ground point, had been inadvertently connected to the power rail. Since the module had multiple ground connections, if the track could be cut, the fault could be overcome without loss of circuit function.

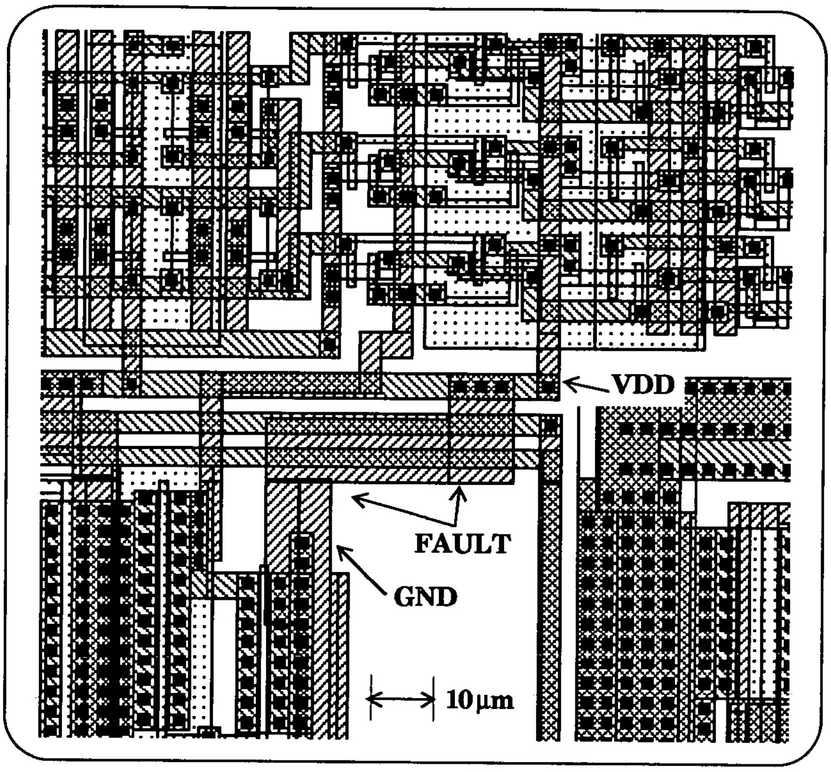


Figure 7.3 Power/Ground Fault

Figure 7.3 shows the position of the power and ground rails, and the offending track. The solution to this problem was provided by a group from the Microelectronics Research Centre at Cambridge University, who could accurately control and focus an ion beam to cut individual tracks. The location of the second layer metal track, and the unused surrounding silicon, made it relatively easy to cut with a low risk of peripheral damage.

Four devices were treated in this manner. The result of this cutting process is shown in figure 7.4. After being connected to a 5V supply and a 12MHz system clock, the current drawn immediately settled to just under 200mA. This verified that the track cutting exercise had been successful.

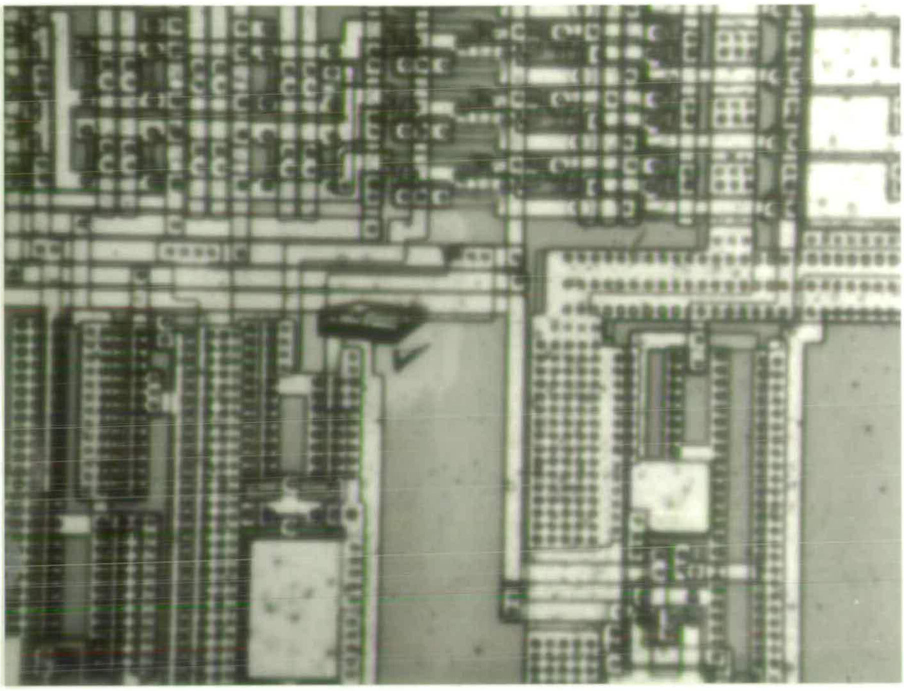


Figure 7.4 Track Cutting

A total of eight devices, consisting of the two power test chips, four cut chips, and two further uncut examples, were each subjected to a variety of functional tests. The following sections will describe the tests carried out, and present the results.

7.3 MEMORY TESTS

After power-up and initialisation, tests to verify the operation of the various memory blocks were run. These routines also verify the microcontroller interface, since all memory accesses involve writing to, and reading from, interface registers on ASIP. The program shown in figure 7.5 performs a basic test of one of the on-chip SRAM blocks; similar routines were used to check the other memory blocks.

```
main() {
  volatile unsigned int i,j,errhi,errlo;
  volatile unsigned int hi,lo;

  serial();                              // Setup serial communications.
  XBYTE[CONTROL]=0;                      // Reset all current ASIC processes.
  printf(“TEST 2 - SIGRAM\n”);           // Print message on terminal.
  errhi=errlo=0;                         // Reset high and low error counts.
  for (j=0; j<=255; j++) {
    XBYTE[ADDHI]=SIGRAM;                 // Select memory page.
    XBYTE[ADDLO]=j;                      // Set memory address register.
    XBYTE[WRITELO]=255-j;                // Write to low byte data register.
    XBYTE[WRITEHI]=255-j;                // Write to high byte data register.
}                                        // Loop for 256 locations.
  for (j=0; j<=255; j++) {
    XBYTE[ADDHI]=SIGRAM;                 // Select memory page.
    XBYTE[ADDLO]=j;                      // Set memory address register.
    hi = XBYTE[READHI];                  // Read from low byte data register.
    if (lo != (255-j)) errlo++;          // If not correct increment error count.
    lo = XBYTE[READLO];                  // Read from high byte data register.
    if (hi != (255-j)) errhi++;          // If not correct increment error count.
  }
  if ((errhi != 0) || (errlo != 0)) {    // If there are any errors then print
    printf("%u lo errors\n",errlo);      // error messages.
    printf("%u hi errors\n",errhi);
  }
```

Figure 7.5 Typical Memory Test Routine

During the first interactive memory checks, it was noted that there were occasional glitches on both the address and data strobes, controlling the transfer and latching of data between the microcontroller emulator and ASIP. These glitches generally appeared coincident with heavy switching of address/data lines on the interface bus. Attempts were made to find the cause of this glitching, including replacing the emulator processor pod, but with no improvement. This glitching problem had also been noted on an unrelated project, using the same emulator. Unfortunately, this noise on the interface was interpreted by ASIP as valid instructions, causing unpredictable results. A partial solution was found by reducing the supply voltage to around 4.7V (from the nominal 5V used initially). Although not totally satisfactory, it did at least allow the functional testing of ASIP to continue. Table 7.2 shows the results for the eight devices used during the initial test phase at 4.7V and 4MHz. Those devices identified by a letter (*i.e.* A, B, D and E), were those which had been powered up without correcting the power supply short, while those with digit codes (*i.e.* 2, 3, 4 and 5), had been processed to correct the fault before being tested.

DEVICE	SIGRAM	RANKRAM	RLUT	LUT	BINRAM
A	Address bit b0 stuck at zero.	PASSED	Low and high bytes stuck at zero	PASSED	PASSED
B	PASSED	Odd locations O.K. Even locations: Low & High bytes = val + 19	Low and high bytes stuck at zero	PASSED	ALL FAILED
D	PASSED	PASSED	Low and high bytes stuck at zero	PASSED	PASSED
E	Even locations O.K. Odd locations: Low byte = value + 19 High byte = value - 5	PASSED	Low and high bytes stuck at zero	PASSED	PASSED
2	ALL FAILED	PASSED	Address bit b0 stuck at one	PASSED	PASSED
3	Even locations O.K. Odd locations: Low byte = value + 19 High byte = value - 3	PASSED	Various high and low byte errors	PASSED	PASSED
4	PASSED	PASSED	Address bit b0 stuck at one	ALL FAILED	ALL FAILED Write enable fault
5	High bytes O.K. Low bytes: Even locations O.K. Odd locations errors	PASSED	Various high and low byte errors	PASSED	PASSED

All tests performed at 4.7V with a 4MHz system clock

Table 7.2 Memory Test Results

The first memory blocks tested were the three, on-chip SRAM blocks. The test procedure involved writing a different value to each memory location, reading back the result, and comparing it with the test data. The errors exhibited by the main 256 x 16 SRAMs (SIGRAM and RANKRAM) on some of the chips did not vary with either voltage or clock frequency. Since these blocks worked on some of the devices tested, it is likely these particular fixed-pattern errors were due to manufacturing faults.

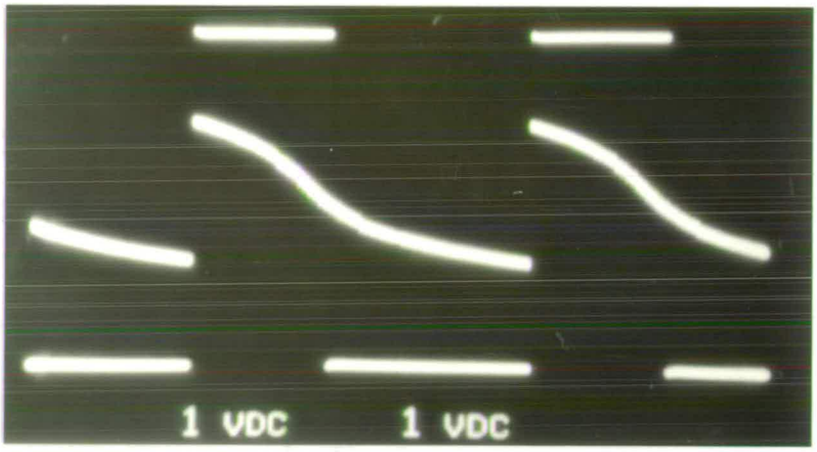
The remaining on-chip SRAM (RLUT), the small 16 x 16 block used during the ranking and polling procedures, failed on all of the tested chips. On the uncut devices all memory locations (high and low bytes) returned zero, while the data read from the remaining chips had some relationship to the memory address. Further simulation of the chip did not reveal any marginal timing or excessive loading which could have caused the non-operation of this memory block.

The two on-chip ROM blocks, which together form the LUT table of address modifiers, were the next part of ASIP tested. The data read from the ROM was compared with the expected results from the ROM compiler. The test was successful for all the chips tested except one. Every LUT location produced the wrong result for this device.

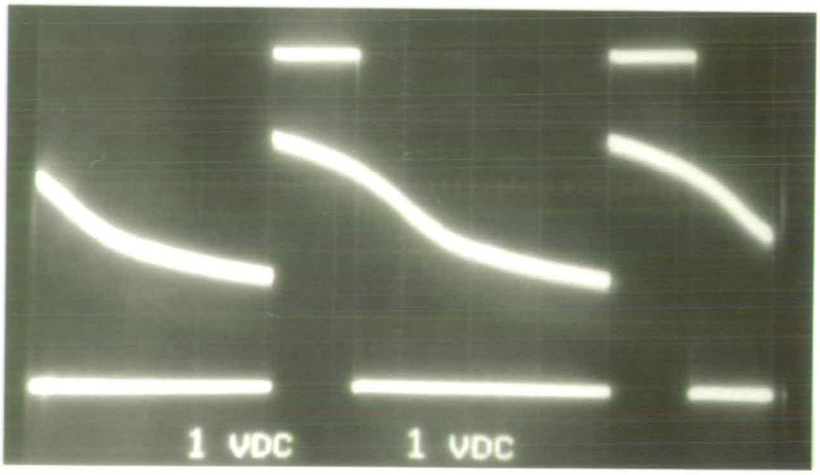
The final memory block, the 8k x 8 binary image store, is situated off-chip but is accessed *via* ASIP, using a similar method to the on-chip blocks. Six of the eight devices passed this test, one failed due to a stuck-at-zero write enable strobe and the other failed with an undiagnosed fault. Other than the total failure of the 16 x 16 SRAM, RLUT, all the memory errors discovered are attributable to processing faults.

7.4 DAC AND COMPARATOR

The input to the DAC can be sourced either from local memory, or from an interface register (BEST), by setting the appropriate bit in the SENTEST register. To test the function of the DAC, a series of values were written to BEST and the response of the DAC measured *via* an analogue output pad on the chip. The comparator, circuit was tested at the same time as the DAC by applying an external voltage to one input of the comparator and the output of the DAC to the other.



(a) Full sweep of DAC output and response of comparator with a 2.4V reference on one input and the DAC output on the other.



(b) Full sweep of DAC output and response of comparator with a 3.1V reference on one input and the DAC output on the other.

Figure 7.6 DAC and Comparator Test Results

The resulting DAC and comparator output responses were monitored using an oscilloscope. The DAC output, and the comparator response to it for two different voltage levels, 2.4V and 3.1V, are shown in figure 7.6 (a) and (b). The results show that the circuits operate as expected.

7.5 SENSOR

The bit-line test mode was selected by writing to the appropriate location in the sensor configuration register, SENTEST. This procedure tests the integrity of the photo-diodes, the connecting bit-lines, sense amplifiers, and associated read-out circuits. One of two complementary test patterns can be selected, each consisting of a series of alternating highs and lows on consecutive bit-lines. The result of the test is a sequence of high and low values on the analogue output of the sensor, CVO.

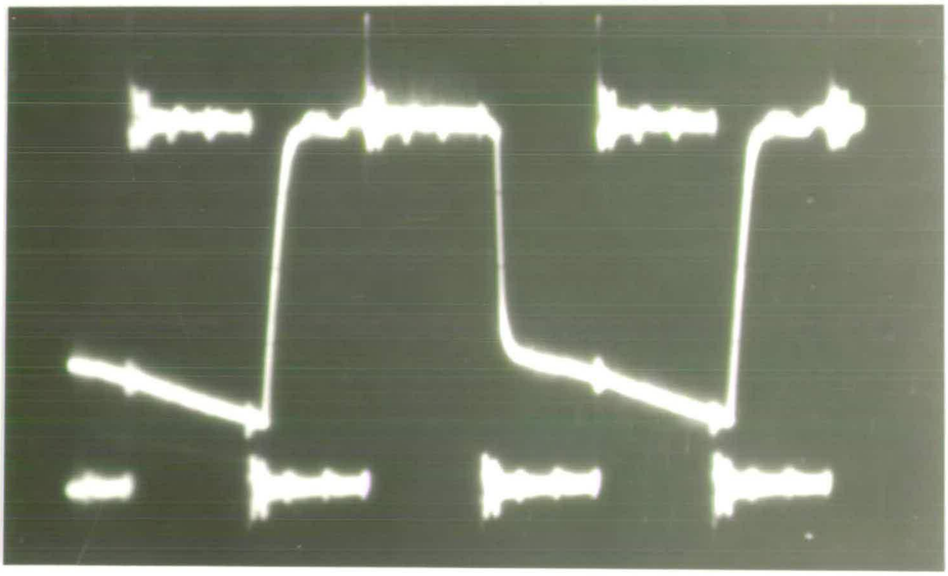


Figure 7.7 Sensor Bit-line Test Output

The image shown in figure 7.7 traces CVO, and the pixel clock used to sample the analogue signal. As can be seen from the results, the output stages of the sensor produce a good '1' but a poor '0'. This is consistent with the HSPICE simulations of the output stages.

The word-line test exercises the vertical shift register and the photodiode array word lines, using an alternating high-low digital pattern. A test-pattern decoder, situated on the opposite side of the array from the vertical shift register, is used to detect this pattern. The output of the decoder is a series of high pulses, which can be monitored on the word line test output pin, WTO. Figure 7.8 traces WTO for one of the test devices running at 0.2 MHz. At higher frequencies, the pulses gradually shrink from nearly 4V, to around 2V at 1MHz. This is due to excessive delays in the combinational decode logic.

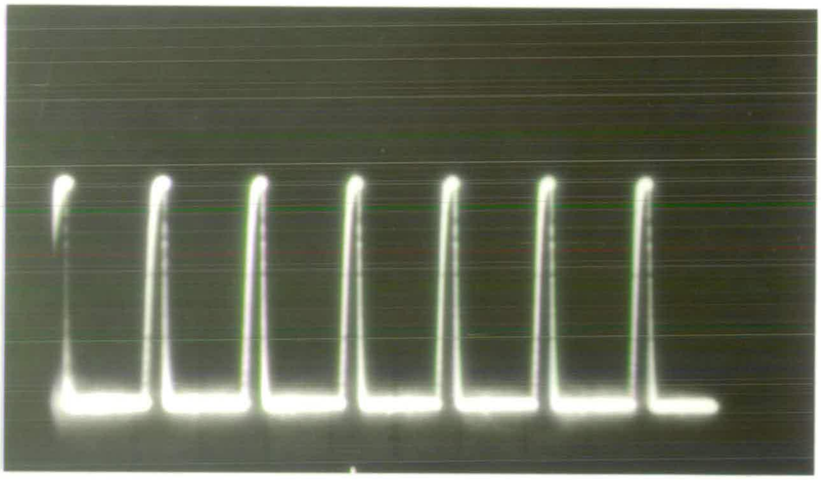
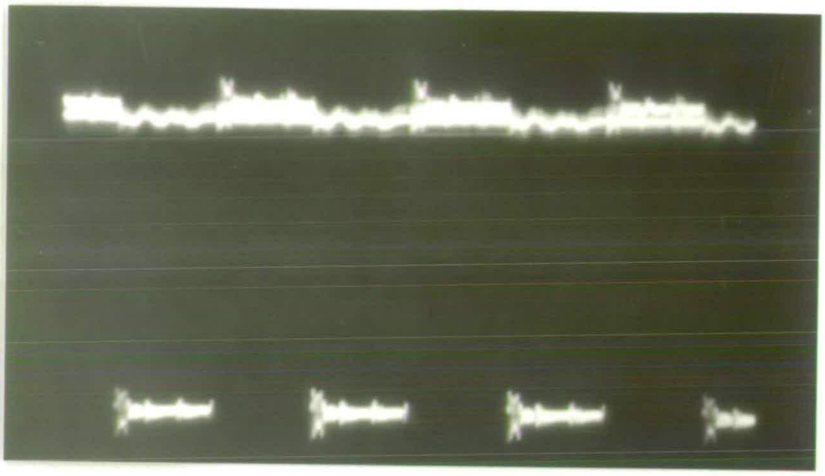
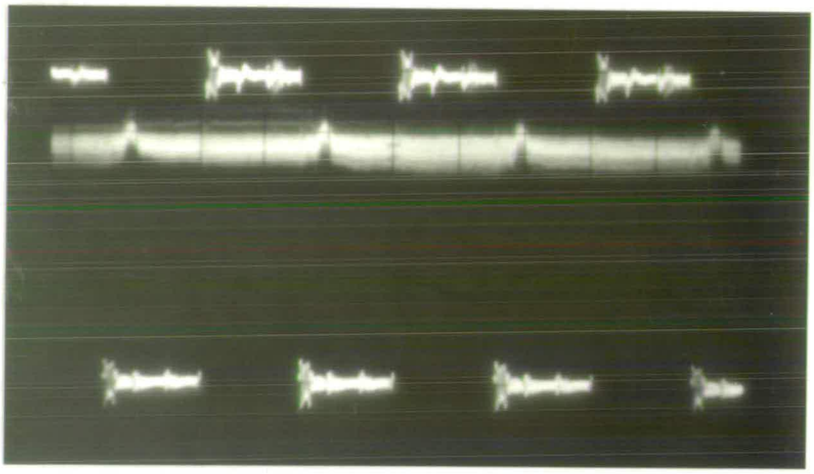


Figure 7.8 Sensor Word-line Test Output

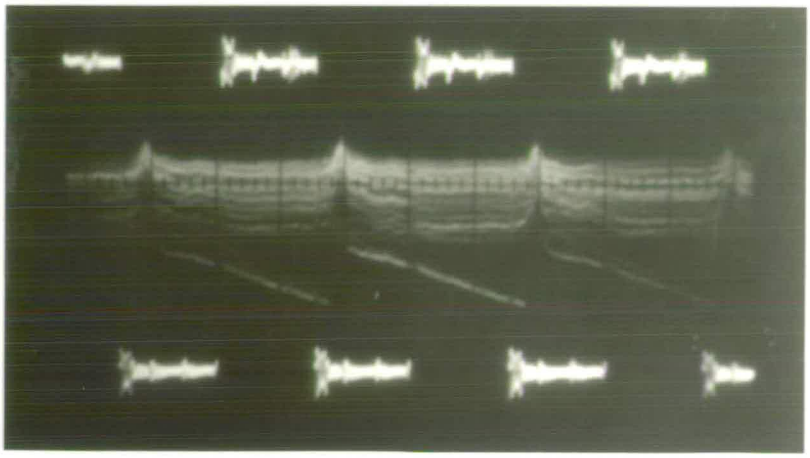
Unfortunately, at this stage another design fault in the full-custom block was discovered. A logic fault in the sensor control logic restricted the sensor so that it could only be operated in test mode. This required further track cutting to allow the sensor to be run in normal mode, but would permanently disable the test modes. This modification was performed on two devices. Preliminary testing of these devices produced encouraging results. Figure 7.9 (a), (b) and (c) show the analogue output of the sensor, CVO, responding to different light levels. Further testing using a focused light source, and a new test rig, incorporating a frame grabber, will be required to fully characterise the operation of the sensor.



(a) Pixel sample clock and sensor output (CVO). No light.



(b) Pixel sample clock and sensor output (CVO). Some light.



(c) Pixel sample clock and sensor output (CVO). Bright light.

Figure 7.9 Sensor Normal Mode Response

The testing of the memory and full custom elements of ASIP has successfully verified their basic operation, but has also of revealed a number of implementation and manufacturing faults. These faults, unfortunately, prevent ASIP being operated as a complete system. The next section describes the testing of the digital preprocessing and image processing logic.

7.6 IMAGE CAPTURE AND PROCESSING.

The remaining tests deal with the verification of the compiled digital logic which implements the preprocessing, correlation, and data analysis functions forming the fingerprint comparison algorithm.

The initial test of the image capture process involved verifying that the process would execute and halt after capturing one image frame. First, appropriate registers for the image capture process were set up, and the process executed by setting the appropriate bit in the control register. Image data is only consumed by the process from the start of a new frame, which is detected by monitoring the frame end strobe. After capturing and processing an image frame, a process complete bit is set in the ASIP's status register. This register is monitored by the software running on the microcontroller.

After verifying the controllability, of the process, the next step was to provide data *via* a multiplexed input, which could source live data from the output of the digitising comparator (*i.e.* sensor data), or synthetic data from an external source. The formatting logic, and address generation logic, is checked by making use of this external input. First, the external binary image RAM is cleared and the process data input is tied to 5V (logic '1'). Then, the process is executed and monitored, *via* the status register, until complete. The contents of the RAM is then examined. Assuming there are no functional (or fabrication) errors, the expected result is FF_{HEX} in every location. The address patterns generated during the execution of the process can also be monitored using a logic analyser attached to the address bus.

This and other, more complex, tests involving the sensor, DAC, and comparator were used to analyse the functionality of the image capture process. These tests verified the basic operation of the process, but also revealed a number of minor logic errors within the design. These errors were traced back to the simulation model, and rectified.

The main elements of the image processing logic; the correlation array, the rank-value filter and the *votepoll* process were checked in a similar fashion. First, each process was checked for controllability, and then internal and external memory was loaded with synthetic data to verify each section.

The correlation controller and the correlation array contained one minor data mapping error, which was not spotted during simulation. The other processes (ranking and polling) revealed no detectible design errors.

7.7 SUMMARY

In conclusion, a prototype single-chip image sensor-processor for fingerprint comparison has been fabricated and tested. Tests were carried out to check the operation of the on-chip memory blocks, RAM and ROM, the DAC and comparator circuit, the image sensor, and all of the digital processing logic.

The results of the memory tests verified that, except for a small cache RAM and errors due to manufacturing yield, all of the on-chip memory worked correctly. A video-rate DAC and comparator circuit forms the core of the image digitisation operator. This module performed as expected, producing results very close to the extracted HSPICE simulations.

Preliminary chip testing exposed a design error in the image sensor output stage: the fault was corrected by ion beam track cutting. In bit- and word-line test modes, which test pixel control and access line integrity, the sensor responded correctly. Further testing of the image sensor will be necessary, using focused light and a new test-rig which incorporates a frame-grabber, in order to fully characterise operation.

Functional testing of the preprocessing logic, *i.e.* image capture and digitisation, confirmed its basic operation. Further testing, using real image data from the sensor, will be necessary to complete the testing of this operator. The image processing logic, *i.e.* correlation, ranking, and polling were individually tested using synthesised data loaded into the on-chip memory. Other than a small number of minor logic errors, the digital processes operated as expected.

The next chapter provides a summary and critical discussion of the work presented in this thesis.

Chapter 8. SUMMARY, DISCUSSION AND CONCLUSIONS

The utilisation of image processing techniques, to solve a wide variety of monitoring and control applications, has been severely limited by the cost of available implementation technologies. The motivation behind the work presented in this thesis, was to find a method to allow the implementation of real-time application specific image processing systems for a few hundred pounds, rather than thousands of pounds as with present methods. In addition to system cost reduction, if the size and power requirements of image processing systems could also be reduced, new markets for low-cost commercial and domestic image processing products would be created, and existing application areas expanded.

8.1 A BRIEF REVIEW

Chapter 2 briefly reviewed the history of implementation methods and technologies for image processing systems, highlighting the limitations of each type, and describing a range of currently available VLSI devices for image processing. VLSI technology and design techniques have enabled the fabrication of massively parallel processing platforms, based on custom VLSI processing elements. Although a large number of different general purpose parallel processing architectures have been developed, no single architecture can efficiently implement all types of image processing function. The proposed solution to the design of efficient, general purpose, image processing systems lies in the development of hybrid multiprocessor computing platforms, combining several different processing architectures in order to allow efficient mapping of the algorithm to the system. This hybrid architecture can also be scaled down to enable the production of application-specific image processing systems based around a small number of high performance custom VLSI devices. A number of image sensors, which provide the image detection front-ends for image processing systems, have also been provided by the use of VLSI technology.

The solution proposed in this thesis, for the low-cost implementation of image processing systems, is to combine the processing power and the image sensing capabilities offered by VLSI on a single, highly integrated device. An image processing system for automated identity verification, based on fingerprint comparison provided a demonstration vehicle for the implementation method. Chapter 3 presented a summary of biometric identification techniques, and commercially available identity verifiers, justifying the choice of a fingerprint comparison system as a technically, and commercially, relevant demonstration system.

Further background regarding the fingerprint comparison system, which forms the basis of the practical work presented in this thesis, was given in chapter 4. An architectural overview of the fingerprint comparison sensor-processor was given in chapter 5, along with a description of the CMOS-based image sensor technology which has enabled the integrated sensor-processor approach to be considered. The remaining chapters described the implementation, fabrication and testing of the demonstration sensor-processor device, ASIP.

8.2 CONCLUSIONS

The objective of the work described in this thesis was to provide a low-cost method for the implementation of application-specific image processing systems. The technical achievements of this work have come very close to reaching that goal. A VLSI device has been fabricated that has succeeded in integrating, or eliminating, through architectural enhancements, nearly all the elements of typical image processing systems *i.e.* image capture, signal conditioning, digitisation, temporary image memory, image preprocessing and analysis logic. In order to build a fingerprint-based identity verifier around this device the only major devices required, other than peripheral system components (*i.e.* keypad, display, fingerprint capture platen, power supply, template store or smart card reader), are a single 64kbit RAM to store the preprocessed binary fingerprint image, and a simple, embedded microcontroller to provide system housekeeping functions. A breakdown of the costs of the major elements of the prototype fingerprint verification system, described in chapter 4, and a similar system based on the ASIP smart sensor-processor is given in table 8.1.

Sub-System	Prototype FVU system	Smart sensor-processor based system
Optics module	20.00	20.00
Image sensor and interface module	170.00	-----[1]
Image processing motherboard[2]	285.00	130.00
Peripheral modules[3]	100.00	85.00
Total component costs[4]:	**£575.00**	**£235.00**

[1] Image sensor now integrated with image processor.

[2] Includes A/D, fast access memory, image processing ASIC(s), microcontroller, reference signature store and support logic. Image processing core based on Parcor1 and Parcor2 for the FVU and on ASIP for the proposed system.

[3] Includes PSU, magnetic card reader, LCD display, box, *etc.*

[4] Component costs only, for 100-off quantities. Does not include assembly costs.

Table 8.1 System Economics

As can be seen, the use of the integrated sensor-processor provides a significant reduction in the component costs from £575 *per* unit to £235. This price does not include assembly costs, which would also be much lower for the ASIP based system due to the reduced number of components.

In addition to the component economics, a system designed using the sensor-processor concept also benefits from the other advantages of VLSI technology, namely small size, low power consumption, and smaller numbers of system components leading to higher system reliability, and very high processing performance. The latter benefit is achieved by efficient mapping of each part of the algorithm to an appropriate processing architecture. Critical to the success of this implementation approach is the ability to successfully integrate the image sensor technology onto the same substrate as the image processor. The CMOS sensor technology developed at Edinburgh University, and used in this work, has numerous architectural design advantages, including choice of array size, pixel aspect ratio, read-out format, and the possibility of closely coupled preprocessing functions, and/or low level image processing. The fact that the CMOS sensor technology is fabricated using the same low cost digital technology as the image processing logic, has an obvious beneficial impact on the system economics. The architectural design freedom offered by the integrated sensor-processor approach, was highlighted by the implementation of the smoothing filter function as part of the image preprocessor in the fingerprint comparison system. In the prototype system, the function was implemented as a pipelined datapath in the digital domain, while the same function was integrated into the read-out logic of the custom sensor architecture at no additional hardware cost.

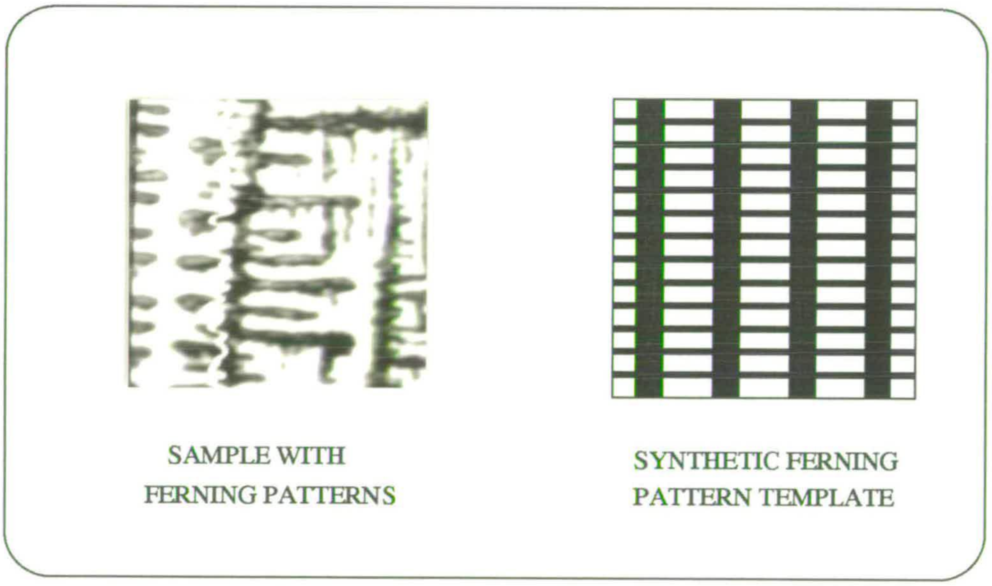
Of course there are disadvantages with this implementation method, many of which became apparent during the development of the demonstration device. The most obvious problem with the implementation method is the risk of a *less than* 100%

correct, first time device. This is due to the tremendous complexity of these sensor-processors, and the inability of existing design tools to provide the necessary mixed-mode and mixed-level simulation and verification required to avoid design errors. This is especially important when integrating compiled and full-custom modules in a single chip.

Post-fabrication testing is also difficult because of the limited number of input and output signals available. Since these sensor-processors are essentially *systems-on-a-chip*, with light in and control signal out, careful consideration has to be given to the accessibility and observability of internal nodes, to allow adequate test coverage. The use of on-chip memory resources can be particularly useful in aiding the testing of buried processes. Synthetic data can be set up in on-chip memory and the process executed. The results can then be read out *via* the memory and checked.

Another criticism of the use of custom VLSI technology as an implementation technology, is the lack of algorithmic flexibility which requires a new design to be created for each application. With the high non-recoverable engineering (NRE) costs of VLSI devices, this is a regarded as significant drawback of the custom implementation approach. This perceived lack of flexibility is not due to the technology itself, but is more likely caused by poor architectural design. With careful system partitioning, the use of a minimal number of additional module parameterisation registers, and high level system configuration and control access through a simple interface, the functions contained in a custom device can be used to solve tasks other than those for which they were originally designed. Further savings can be made through the amortisation of the NRE costs across a number of different designs, by the use of macro or library cell blocks. This enables cells such as the sensor, DAC, and compiled blocks such as the correlation array to be re-used in future designs.

The architectures used to implement the image processing functions forming the fingerprint comparison algorithm, were also used to demonstrate the feasibility of a proposed medical pattern matching system for fertility testing being developed at the University of Massachusetts[33]. The method relies on determining the presence, or not, of a particular hormone produced during the female menstrual cycle. Samples taken from women are prepared, and analysed for the existence of crystallisation patterns, known as 'ferning' patterns, which are an indicator of the presence of a particular hormone. In the fingerprint comparison system a template produced from an original fingerprint is compared with a presented image for verification of identity. In this second application, the comparison template was generalised representation of a ferning pattern. An example ferning pattern and comparison template are shown in figure 8.1.



SAMPLE WITH FERNING PATTERNS

SYNTHETIC FERNING PATTERN TEMPLATE

Figure 8.1 Ferning Pattern Sample and Template

Using the image pre-processing functions and the correlation elements of the fingerprint system it was shown that it was possible to determine the presence of ferning patterns in the test images. Only minor changes to the adaptive threshold function and different high level data analysis processing were required to change the fingerprint comparison system into a fertility monitor. This example demonstrates the flexibility of the implementation approach.

8.3 FUTURE RESEARCH

As discussed earlier, the first prototypes of fingerprint comparison smart sensor-processor devices failed to operate as specified. One of the main difficulties was the massive complexity of the design, and the fact that it combined analogue and digital parts. The tools used to implement the design were not ideal for the task, with no means of verifying of the integration of the full custom analogue and compiled digital elements of the design. This led to the introduction of several design errors. More sophisticated CAD tools for the simulation, implementation and verification of mixed technology (analogue, digital) and method (full-custom, compiled) devices are becoming available. There are two distinct design flows required for analogue and digital circuit design. For successful mixed signal circuit design these tools will need to provide a number of additional features. These include system level simulation, behavioural modelling to allow analogue cells to be represented in the digital design environment and circuit extraction to allow implementation verification of full custom circuits against behavioural and functional models. Additional features such as synthesis automatic test pattern generation tools for digital logic and interactive floorplanning tools are also now available to the designer. All of these tools should enable complex designs, like the fingerprint comparison device, to be developed with a greater degree of confidence in *right-first-time* silicon.

Another way of reducing the design risk during the production of a prototype or limited production system, is to partition the proposed system into more than one device at appropriate algorithmic breakpoints. An appropriate algorithmic breakpoint is one where a function cannot start until another has completed processing. For the fingerprint algorithm such a breakpoint occurs after the thresholding function has stored the binary fingerprint data in memory prior to the start of the comparison process. For a future redesign of the fingerprint comparison device, the architecture could be split between the image sensing/preprocessing and image analysis parts of the system. This would result in a smart sensor-processor which integrates the sensor, smoothing function, digitisation, and adaptive thresholding elements of the function, resulting in a binary representation of a fingerprint image. A second, purely digital device, would then implement all of the image comparison and analysis functions using the binary fingerprint and a signature template as input data. Although the verification of the two device approach would be slightly simpler the disadvantage is that the total prototyping costs would be significantly higher.

The limiting factor in the design of low-cost image processing systems has been the cost of the processing hardware required to execute image processing functions in real-time. As VLSI technologies advance, with smaller geometries and higher levels of integration, the price of processing power is continuing to fall dramatically, enabling the production of much lower cost systems with ever greater capabilities. Soon, however, the limiting factor will not be the cost of the silicon which will dominate, but the cost of post-fabrication testing and device packaging. Testing the analogue portions of mixed analog-digital devices is particularly time consuming. Built-in self test (BIST) and other novel test techniques, such as those used to test the CMOS sensor technology as a digital device, will have to be developed for analogue circuits.

Packaging costs form a significant proportion of the manufacturing cost of an

ASIC. This is particularly true for image sensor devices with their optical requirements. New packaging technologies, such as flip-chip, TAB bonding, and multi-chip modules, are being developed which promise to deliver higher levels of system integration, reduced system size and, eventually, lower cost. Using the multi-chip module approach the whole of the fingerprint comparison system (excluding peripheral human interface devices) could be integrated into a package about 35 mm square (see figure 8.2).

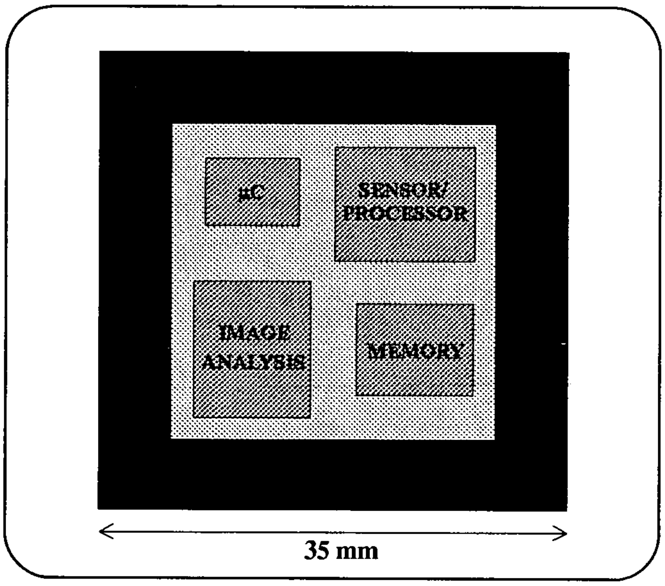


Figure 8.2 MCM Fingerprint Comparison System

In conclusion, VLSI smart sensor-processor architectures are an appropriate method for the implementation of image processing systems. This has been demonstrated by the successful implementation of a fingerprint comparison system as the single-chip, highly integrated sensor-processor presented in this thesis.

REFERENCES

Biometrics

1 Braggins, D., "Reading Between the Pixels", Image Processing, pp. 43-45, May/June 1991.

2 Driscoll, E.C. and Fowler, R.C., "A Comparison of Centralized versus Distributed Architectures in Biometric Access Control", Proc. 1989 Int. Carnahan Conference on Security Technology, Zurich, pp. 193-198, Oct. 1989.

3 Harman, L.D., Khan, M.K., Lash, R., Kamig, P.F., "Machine Identification of Human Faces", Pattern Recognition, Vol. 13, Iss. 2, pp. 148-153, Jan. 1991.

4 Frazer, F.,"Thumbs Up for Security", Scotsman newspaper, Edinburgh, p. 28, 26th Nov.1991.

5 Holmes, J.P., Maxwell, R.L. and Wright, L.J., "A Performance Evaluation of Personnel Identification Devices", Sandia National Laboratories, Albuquerque, 1990.

6 Hughes, P.A., Green, A.D.P., "The Use of Neural Networks for Fingerprint Classification", Proc. 2nd I.E.E. Int. Conf. on Artificial Neural networks, Bournemouth, U.K., Nov. 1991.

7 MacGregor, P. and Welford, R., "Veincheck: Imaging for Security and Personnel Identification", Advanced Imaging, Vol. 6, No. 7, pp. 52-54, July 1991.

8 McCorquodale, "Smart cards: major new initiatives", Electronic Banking & Finance, pp. 3-4, May 1990.

9 Maxwell, R.L. and Wright, L.J., "A Performance Evaluation of Personnel Identity Verifiers", Sandia National Laboratories, Albuquerque, 1987.

10 Maxwell, R.L., "The Status of Personnel Identity Verifiers", Sandia National Laboratories, 1985.

11 Miller, B.L., "1990 Biometric Industry Directory", PIN, Warfel & Miller, Inc., 1990.

12 Miller, B.L., "Biometric Industry Update: Part 1", PIN, Warfel & Miller, Inc., pp.4-7, Sept. 1991.

13 Miller, B.L., "Biometric Industry Update: Part 2", PIN, Warfel & Miller, Inc., pp.1,5-7, Oct. 1991.

14 Parks, J.R., "BIOMETRICS - Facts, Fantasies and Futures", Biometrics Conference, Elsevier,1989.

15 Parks, J.R., "Biometrics: The People Sensors", Sensor Review, pp. 79-84, April 1989.

16 Peckham, J., "I'd Know That Voice Anywhere (Security)", Systems International, Vol. 17, Iss. 9, pp. 49-50, Sept. 1990.

17 Price, W.L., Chorley, B.J., "The Intelligent Token or 'Super-Smart' Card", Smart Card: The Future of IC Cards, Proc. IFIP Int. Conference, Laxenburg, Austria, pp. 133-138, Oct. 1987.

18 Stonham, T.J., "Practical Face Recognition and Verification with Wisard", Aspects of Face Processing, pp. 426-441, Martinus-Nijhoff, 1986.

19 Suenaga, Y., Watanabe, Y.,"A Method for the Synchronized Acquisition of Cylindrical Range and Colour Data", IEICE Transactions, Vol. E 74, No. 10, Oct. 1991.

20 Sutherland, K., Renshaw, D.A., Denyer, P.B., "A Novel Automatic Face Recognition Algorithm Employing Vector Quantisation", Proc. IEE Colloqium on Facial Recognition and Storage, London, Jan. 1992.

21 Yoshimura, M., Kato, Y., Matsuda, S., Yashimura, I., "On-line Signature Verification Incorporating the Direction of Pen Movement", IEICE Transactions, Vol. E 74, No. 7, July 1991.

Image Processing - Applications, Architectures and Systems

22 Bruce, W. H., "Development and Implementation of Fingerprint Matching and Recognition Algorithms", BSc Project Report HSP 436, Dept. Electrical Engineering, University of Edinburgh, 1985.

23 Bruce, W.H., "Microcontroller Software for Fingerprint Verification Unit", Appendix, ISG Internal Report, Finger-22-B, Dept. of Electrical Engineering, Edinburgh University, Aug. 1989.

24 Bruce, W.H., "Fingerprint Matching and Recognition Algorithm", Unpublished

Ph.D. Thesis, Dept. Electrical Engineering, University of Edinburgh.

25 Chakrabarti, C. and JaJa, J., "VLSI architecture for Template Matching", Proc. IEEE ISCAS, pp. 69-72, 1990.

26 Denyer, P. B., "Skin Pattern Recognition Method", ISG internal report Finger-001-C, Dept. Electrical Engineering, University of Edinburgh, 1988.

27 Flynn, M., "Some Computer Organisations and their Effectiveness", IEEE Trans. Comp., Vol. C-21, No. 9, pp. 948-960, 1972.

28 Graham, M.D., "The diff4: A Second-Generation Slide Analyser", Computing Structures for Image Processing, pp. 179-193, ed. Duff, Academic Press, 1983.

29 Groen, F.C.A., Jonker, P.P., Duin, R.P.W., "Hardware versus Software Implementations of Fast Image Processing Algorithms", NATO ASI series, Vol. F42 Real-time Object Measurement and Classification, Springer-Verlag, 1988.

30 Hodgkiss, D., "Hard and Soft Options", Image Processing, pp. 10-12, July/Aug. 1992.

31 Hogan, B., "High-Performance Image Processing on a Massively Parallel Computer", Advanced Imaging, pp. 42-48 & 85, Oct. 1990.

32 Howington, L.C., "Automated License Plate Reading", Advanced Imaging, pp. 46-49, Sept. 1989.

33 Jackson, G.S. and Ciesielski, M.J., University of Massachusetts at Amherst, private communication, Nov. 1991.

34 Jones, R.E., "Vision Helps Canned Guinness Keep Ahead", Advanced Imaging, pp. 14-16.

35 Kurokawa, H., Matsumoto, K., Temma, T., Iwashita, M., Nukiyama, T., "The Architecture and Performance of Image Pipeline Processor", Proc. VLSI '93, pp. 275-285, 1983.

36 Lee, C.Y., Catthoor, F., De Man, H., "Real-Time Regularity Detection for Robot Vision Using a Customized Architectural Approach", Proc. IEEE ISCS, pp. 1493-1496, 1990.

37 O'Brien, K. and D. O'Conchuir, "Real Time Image Processor for In-line Robotic Control in Flexible Manufacturing Systems", Proc. IEEE IPCCC, pp. 788-794, Phoenix, 1990.

38 Potter, J.L., "The Massively Parallel Processor", MIT Press, 1985.

39 Reddy, D.R. and Hon, R.W., "Computer Architectures for Vision", Proc. Image Understanding Workshop, Science Applications, pp. 169-185, 1979.

40 Ruttlidge, H., "The Performance of an Automatic Metaphase Finder used for Chromosome Aberration Scoring", Institute of Physics Conference Series, No. 44, pp. 210-219, 1975.

41 Sanz, J.L.C., "Which Parallel Architectures are Useful/Useless for Vision Algorithms?", Machine Vision and Applications, pp. 167-173, Springer-Verlag, 1989.

42 Souza, L.A., Piedade, S.M., Caeiro, J.J., "A High Performance Image Processing System", Proc. IEEE ISCS Conf., pp. 751-754, 1990.

43 Stix, G., "Check it out", Scientific American, pp. 98-99, Aug. 1992.

44 Tucker, J.H. and Shippey, G., "Basic Performance Tests on the CERVIFIP Linear Array Prescreener", Analytical and Quantitative Cytology, Vol. 5, No. 2, pp. 129-137., 1983.

45 Vellacot, O.R., "The Imputer - A Niche Architecture in Electronic Vision", JFIT Technical Conf., Keele, March 1993.

46 Williams, S., Bruce, W.H., "FVU Hardware & Software Description", Internal Report, Dept. Electrical Engineering, University of Edinburgh 1989.

VLSI for Image Processing

47 Anderson, S.,"FVU ASICs - Data Sheets", ISG internal report Finger-20-A, Dept. Electrical Engineering, University of Edinburgh, Feb. 1989.

48 Anderson, S.,"FVU ASICs - Design Data", ISG internal report Finger-21-A, Dept. Electrical Engineering, University of Edinburgh, June 1988.

49 Arvind, D.K., Robinson, I.N., Parker, I.N., "A VLSI Chip for Real-Time Image Processing", Proc.IEEE ISCS Conf., pp. 405-408, Newport Beach, May 1983.

50 Batali, J., "A Vision Chip", M.Sc. Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1981.

51 Beynon, J.D.E and Lamb, D.R., “Charged-Coupled Devices and their Applications”, Chapt. 4, ISBN 0-07-084522-0, Pub. McGraw-Hill, 1980.

52 Blanz, W.E.,”VLSI-oriented Architectures for Real-time Image Processing”, Proc. of the SPIE - The International Society for Optical Engineering, Vol. 1246, pp. 57-68, Feb. 1990.

53 Brofferio, S., Monti, M., Rampa, V., Taliercio, M., “VLSI Recursive Motion Estimator Chip Set”, Proc. IEEE ISCS, pp. 57-60, 1990.

54 Catthoor, F., O’Gorman, L., Jain, R., “An ASIC Architecture for Contour Line Filtering”, IEEE ICASSP, pp. 1056-1059, 1988.

55 Corry, A. and Patel, K., “Architecture of A CMOS Correlator”, The GEC Journal of Research, Vol. 1, No. 1, pp. 35-38, 1983.

56 Goudet, C., Mathieu, Y., Concordel, G., Demassieux, N., “A Real-Time Multi-Kernel Picture Convolver”, Proc. IEEE CICC, pp. 17.8.1 - 17.8.4, Boston, 1990.

57 Hesketh, J.A. and Burrows, D.F., “A Comparison of Methods for Large-scale Mixed Signal ASIC Design”, Unpublished paper, LSI Logic Europe Plc, Bracknell, U.K., June 1992.

58 Jutand, F., Demassieux, N., Artieri, A., “A New VLSI Architecture for Large Kernel Real Time Convolution”, Proc. IEEE ICASSP, pp.921-924, 1990

59 Lea, R.M., “SCAPE: A Single Chip Array Processing Element for Image Analysis”, Proc. VLSI Conf., pp. 285-294, 1983.

60 Legat, J-D. and De Muelenaere, P.,“A High Performance SIMD Processor for Binary Image Processing”, Proc. IEEE Custom Integrated Circuits Conf., pp. 17.4.1-17.4.4, Boston, 1990.

61 Maguire, G., O’Brien, J., Finnerty, M., Daly, J., Mitchell, S., Rida, N., O’Riordan, J., Griffin, F., “IRIS: A 20MHz Image Recognition System”, Proc. IEEE Custom Integrated Circuits Conf., pp. 17.6.1-17.6.4, Boston, 1990.

62 Maguire, P., Griffin, F., O’Sullivan, M., O’Brien, J., “OPTIC - A high performance image processor”, Proc. 18th ESSCIRC Conf., pp. 143-146., Copenhagen, 1992.

63 Maruyama, M., Nakahiri, H., Araki, T., Sakiyama, S., Kitao, Y., Aono, K., Yamada, H., “An Image Multiprocessor on a Single Chip”, IEEE Journal of

Solid State Circuits, Vol. 25, No. 6, Dec. 1990.

64 Quenot, G. and Zavidovique, B, "A Data-Flow Processor for Real-Time Low-Level Image Processing", Proc. IEEE Custom Integrated Circuits Conf., pp. 12.4.1-12.4.4, San Diego, 1991.

65 Ramaswamy, R., Brebner, G., Aspinall, D., "Architecture of a Highly Reliable Systolic Correlator Array", Proc. IEEE ICASSP Conf., pp. 917-920, 1990.

66 Rauscher, R.H. and Maeder, A.J., "A Chip For Real-Time Skeleting of Images", Proc. IEEE Custom Integrated Circuits Conf., pp. 15.4.1-15.4.4, Boston, 1990.

67 Reder, J.R., "VLSI Implementation of Multiple Binomial Windows for Real-time Image Processing", Proc. IEEE ISCS Conf., pp. 739-742, 1990.

68 Ruetz, P.A. and Brodersen, R.W., "Architectures and Design Techniques for Real-Time Image-Processing ICs", IEEE Journal of Solid State Circuits, Vol. 22, No. 2, April 1987.

69 Ruetz, P.A., "The Architectures and design of a 20-MHz Real-Time DSP Chip Set", IEEE Journal of Solid State Circuits, Vol. 24, No. 2, April 1989.

70 Shah, I., Akiwumi-Assani, O., Johnson, B., "A Chip-Set for Lossless Image Compression", 1990 Proc. IEEE Custom Integrated Circuits Conf., pp. 17.7.1-17.7.4, Boston, 1990.

71 Wang, G. and Lu, M., "3 x 3 Filter", Dept. of Electrical Engineering, Edinburgh University, personal communication, Dec. 1989.

Sensors & Sensor-Processors

72 Anderson, S., Bruce, W.H., Denyer, P.B., Renshaw, D., Wang, G., "A Single Chip Sensor and Image Processor for Fingerprint Verification", 1991 Proc. IEEE Custom Integrated Circuits Conf., San Diego, pp. 12.1.1-12.1.4, 1991.

73 Bernard, T.M., Zavidovique, B., Devos, F., "The NCP Retina: An Imager, a Halftoner and a Micro-Grained Array Processor on the Same Chip", 18th ESSCIRC Conf., pp. 1559-162, Sept. 1992.

74 Chen, K., Afgahi, M., Danielsson, P.E., Svensson, C., "PASIC: A Processor-A/

D converter-Sensor Integrated Circuit", Proc. IEEE ISCS Conf., New Orleans, pp. 1705-1708, 1990.

75 Denyer, P.B., Renshaw, D., Wang, G., Lu, M., Anderson, S., "On-chip CMOS Sensors for VLSI Imaging Systems", Proc. VLSI 91, Edinburgh, 1991.

76 Forchheimer, R., Chen, K., Svensson, C., Odmark, A., "Single-Chip Image Sensors with a Digital Processor Array", Submitted to Journal of Solid-State Circuits, May 1990.

77 Hakkarainen, J.M. and Lee, H.-S., "A 40x40 CCD/CMOS AVD Processor for use in a Stereo Vision System", 18th ESSCIRC Conf., pp. 151-154, Sept. 1992.

78 Jansson, C., Ingelhag, P., Svensson, C., Forchheimer, R., "An addressable 256x256 Photodiode Image Sensor Array with an 8-bit Digital Output", 18th ESSCIRC Conf., pp. 151-154, Sept. 1992.

79 Kemeny, S., Torby, H., Meadows, H., Bredthauer, R., LaShell, M., Fossum, E.R., "CCD Image Sensor with Differential, Pyramidal Output for Lossless Image Compression", 1991 Proc. IEEE Custom Integrated Circuits Conf., San Diego, pp 12.6.1-12.6.4, 1991.

80 Kioi, K., Shinozaki, T., Toyoyama, S., Shirikawa, K., Ohtake, K., Tsuchimoto, S., "Design and Implementation of a 3D-LSI Image Sensing Processor", submitted for publication in the IEEE Journal of Solid-State Circuits, 1991.

81 Lyon, R.F., "The Optical Mouse and an Architectural Methodology for Smart Digital Sensors", VLSI systems and computations, pp. 1-19, Springer-Verlag, 1981.

82 Middelhoek, S. and Audet, S.A., "Silicon Sensors", Academic Press, 1989.

83 Mead, C., "Analog VLSI and Neural Systems", Addison-Wesley, 1989.

84 Renshaw, D.A., Denyer, P.B., Wang, G., Anderson, S., "Application Specific Vision Sensor Processors", report to SERC, contract GR/F 36538, July 1992.

85 Renshaw, D.A., Denyer, P.B., Wang, G., Lu, M., "ASIC Vision", 1990 Proc. IEEE Custom Integrated Circuits Conf., Boston, pp. 3038-3041, 1990.

86 Renshaw, D.A., Denyer, P.B., Lu, M., Wang, G., "A Single-Chip Camera with Automatic Exposure Control", IEEE ISIC, pp. 346-359, Sept. 1991.

87 Renshaw, D.A. and Denyer, P.B., "ASIC Image Sensors", 1990 Proc. IEEE

International Symposium on Circuits and Systems, pp. 7.3.1-7.3.4, 1990.

88 Tanner, J.E. and Mead, C., "A Correlating Optical Motion Detector", Conf. on advanced Research in VLSI, MIT, pp. 57-64, 1984.

89 Shimmi, T., Kobayashi, H., Yagi, T., Sawaji, T., Matsumoto, T., Abidi, A.A., "A Parallel Analog CMOS Signal Processor for Image Contrast Enhancement", 18th ESSCIRC Conf., Copenhagen, pp. 163-166, Sept. 1992.

90 Wang, G., Denyer, P.B., Renshaw, D., Lu, M., "A Simple Solution for Image Sensor Testing", IEE ISIC, pp. 202-205, Sept. 1991.

91 Wang, G., Renshaw, D., Denyer, P.B., Lu., M., "CMOS Video Cameras", Proc. EuroASIC 91, pp. 100-104, 1991.

92 Yu, P.C., Lee, H-S., "A CMOS Resistive-fuse Processor for 2-D Image Acquisition, Smoothing, and Segmentation", 18th ESSCIRC Conf., Copenhagen, pp. 147-150, Sept. 1992.

Other References

93 Allen, P.E., Holberg, D.R., "CMOS Analog Circuit Design", pp. 333-348, HRW, 1987.

94 Cheng, E.K., Mazor, S., "The Genesil Silicon Compiler", Silicon Compilation, Chapt. 9, pp. 361-405, Ed. Gajski, D.D., Addison-Wesley, 1988.

95 ES2, "SOLO 1400 Overview", Pub. ES2 Ltd., Bracknell, U.K., Jan. 1990.

96 Hamilton, A. and Brownlow, M., "Full-Custom Layout Using Magic and SOLO", internal document, Dept. of Electrical Engineering, University of Edinburgh, July 1989.

97 Hamilton, A., Baxter, D.J., Churcher, S., "How to Import Magic CIF Files", internal document, Dept. of Electrical Engineering, University of Edinburgh, June 1990.

98 Hitex, "Teletest 51 User's Manual", Pub. Hitex-Systementwicklung, Karlsruhe, Germany, June 1989.

99 Keil, "C51 Compiler User's Guide", Pub. Keil Electronik GmbH., Munich, Germany, 1989.

100 Meta-Software, “Hspice User’s Manual”, Pub. Meta-Software, Inc., Campbell, California, 1990/91.

101 Murray, A.F., “Specification of an Optical Front-End for Fingerprint Sensing”, ISG Report No. Finger-005-B, Dept. Electrical Engineering, University of Edinburgh, Jan. 1988.

102 Ousterhout, J.K., “Magic: A VLSI Layout System”, Proc. of 21st Design Automation Conf., *pp.* 152-159, 1984.

103 Scott, W.S., Mayo, R.N., Hamachi, G., Ousterhout, J.K., “1986 VLSI Tools: Still More Works by the Original Artists”, Report No. UCB/CSD 86/272, EECS, University of California, Berkeley, California, Dec. 1985

104 Silicon Compiler Systems, “Genesil Designer”, Pub. Silicon Compiler Systems, Corp., San Jose, California, 1987.

Appendix A. PUBLICATIONS

REPRINTED FROM

Proceedings of the IEEE 1991 CUSTOM INTEGRATED CIRCUITS CONFERENCE

Town & Country Hotel
San Diego, California

May 12-15, 1991

Reprinted from PROCEEDINGS OF THE IEEE 1991 CUSTOM INTEGRATED
CIRCUITS CONFERENCE, San Diego, California, May 12-15, 1991

A SINGLE CHIP SENSOR & IMAGE PROCESSOR FOR FINGERPRINT VERIFICATION

S. Anderson, W. H. Bruce, P. B. Denyer, D. Renshaw & G. Wang

Department of Electrical Engineering, University of Edinburgh
The King's Buildings, Edinburgh, EH9 3JL, Scotland, UK

ABSTRACT

This paper describes a real-time image processing system, including an image sensing array, which is implemented as a single VLSI device. The architectural overview and system performance figures demonstrate that, through efficient integration of the sensing, processing and memory elements of the image-processing system, increased performance and greatly reduced manufacturing costs can be achieved for a specific application (in this case fingerprint verification). The device, designed and fabricated using a standard 1.2µm ASIC CMOS process and 18MHz operation, includes a 258 x 258 image sensor array, real-time image normalisation and image matching circuitry, 8K bits of fast access memory and 16K bits of ROM.

INTRODUCTION

Integrated sensor-processor technology provides a new method for implementing image processing systems IPS as single chips. Typical IPS for the analysis and recognition of images conventionally consist of five main elements :- sensor, data preprocessor, image analysis hardware, memory and CPU. The input medium to this class of system is visible electromagnetic radiation which is sensed and converted into an electrical representation of the image. The sensor can be based on vidicon or solid-state CCD or MOS technology. The electrical signal usually corresponds to some standard such as PAL format video in order to simplify the interfacing to various systems, or it may simply be a raster scan read-out. A preprocessor stage then performs global functions such as spatial noise filtering, A/D conversion and image normalisation before the main image processing task is carried out. The core of the system can be either a high-performance single-chip DSP (e.g. TMS320) or, for real-time operation, an array of general purpose processors (e.g. Transputers) or dedicated image array-processors (e.g. IMS A110). Memory is required to buffer images, store reference data and results for post-analysis. The final component of a generic IPS is a low-bandwidth (when compared with the image processor) microprocessor which is needed to perform system control functions and analysis of results.

A prototype system for personal verification based on fingerprint matching which contains all the elements of such a typical image processing system, *(Figure 1)*, has been developed at Edinburgh University[1]. The sensing unit used was a CCD based camera module. The preprocessing functions and the main comparison algorithm were implemented as a two device ASIC chip-set with a microcontroller performing all other system functions. The system has also been integrated as a single sensor-processor chip *(Figure 1)* as described below.

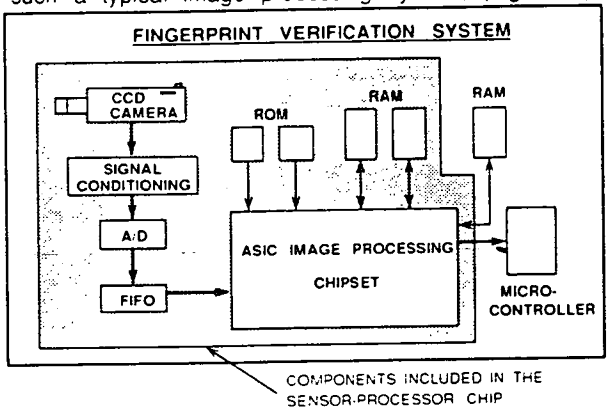

FINGERPRINT VERIFICATION SYSTEM
CCD CAMERA
SIGNAL CONDITIONING
A/D
FIFO
ROM
RAM
RAM
ASIC IMAGE PROCESSING CHIPSET
MICRO-CONTROLLER
COMPONENTS INCLUDED IN THE SENSOR-PROCESSOR CHIP

Figure 1: Block diagram of fingerprint matching system

ALGORITHM

A simple optical system produces a high-contrast image of a fingerprint by utilising the total internal reflection effect of a prism. The image is focused onto the sensor which produces an analogue representation of the data. An image preprocessing stage filters and adaptively thresholds the grey scale image to produce

CH2994-2/91/0000/0061 $1.00 © 1991 IEEE

a binary image ready for the matching stage. Basically the comparison is then performed as follows: to accommodate the translational and rotational movement the comparison algorithm correlates a number of reference signatures against the fingerprint over a range of angles and offsets. Matching decisions are then taken on the basis of results in this multi-dimensional correlation space. This is a reliable but computationally intensive process. To verify a fingerprint against a known reference in a time-scale of 1 second requires the equivalent of $2x10^9$ arithmetic operations per second. The results of these correlations are tabulated and compared against a specified threshold and a decision is made whether or not the fingerprint presented is a good match. The architecture used to implement the system as a single device is presented in this paper and its performance compared with that of the prototype, highlighting the advantages of integrating image sensing and processing on the same silicon substrate.

ARCHITECTURE : SENSOR

Integration of the image sensor has been made possible by the recent development of a high quality, customisable sensor array designed to be fabricated using the same standard low-cost ASIC CMOS process as is used for the surrounding digital image-processing circuitry. The sensing technology [2],[3] is based on a photodiode array and includes all necessary amplification and control circuitry on-chip. Middelhoek and Audet [4] state that one of the important disadvantages of integrated smart sensors is that the process required for the production of the sensing elements is generally not compatible with the process required for the digital circuitry. In the case of image processing we have now demonstrated that a more than adequate sensor performance is achieved without compromising the requirements for the rest of the system.

The nature of the sensor technology has allowed the sensor array to be customised to the requirements of the application. In this case a 258 x 258 array with a pixel aspect ratio of 4:3 has been chosen. Raster read out from this performs a simple spatial transform of the image to correct for optical distortion of the print image from the angled prism face. The clocking requirement is a simple single phase clock and power is provided by the same 5v supply that is used by the surrounding digital logic. The addressing and read out circuits have been modified to allow local 2-D smoothing to take place as the image data is scanned out. This is performed in the analogue domain and implements the required low-pass filtering without any hardware overhead, an example of the benefits of being able to tailor the sensor technology to the application.

ARCHITECTURE : PROCESSOR

The output signal from the sensor array is an analogue representation of the image at each pixel site. This signal is processed to give a normalised binary (i.e. black and white) representation of the fingerprint image which can then be interpreted by the digital processing logic. An adaptive threshold technique is employed to eliminate gross intensity variation across the image caused by uneven illumination of the fingerprint and variable skin characteristics. *Figure 2* shows two examples of grey-level fingerprints before and after the adaptive thresholding function has been applied. The image is divided into 256 16 x 16 pixel patches and local thresholds are calculated for each patch using a successive approximation method. The thresholds are then applied to the image to give the desired ratio of black to white pixels (usually 50%). The architecture implemented to achieve this function is described in greater detail in the next section. An off-chip 64K bit static RAM is used to store the resulting binary image. The data is then ready for correlation with the reference fingerprint data which has been pre-loaded into local memory.

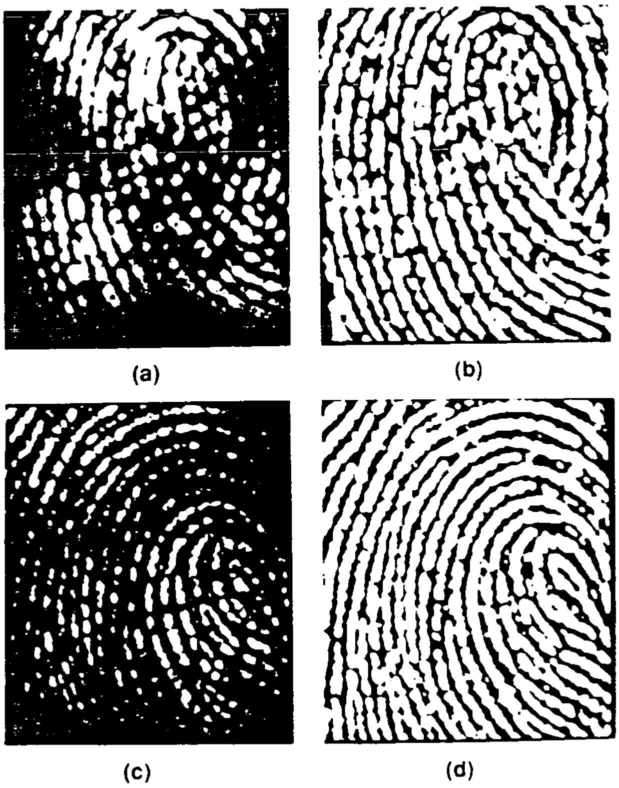


Figure 2: Grey-level fingerprint images (a),(c)
Normalised binary fingerprint images (b),(d)

The matching process is performed by an 8 x 8 matrix of correlation cells and a complex controller/address generator. The controller ensures the flow of data into the parallel correlation array is maximised to achieve the necessary high throughput. Fast on-chip memory close to the array together with pre-calculated address modifiers (to accommodate rotation of the image) allows 64 single-bit comparisons to be made on every cycle of the 18MHz system clocks. Each cell counts pixel matches and performs comparisons with locally held reference scores. At appropriate moments during the correlation sequence the current values of the 64 score registers are read and ranked in result tables for analysis after all the correlations have been completed. The final stage of the process is to analyse the ranked scores to determine whether or not the images match. Circuitry is provided on-chip to perform this function. *Table 1* provides a summary of the main features provided by the single-chip sensor processor.

PERFORMANCE

To illustrate the advantages offered by integrated sensor-processor systems, this section looks in detail at the digitisation and normalisation function of the fingerprint matching system. Assume the image size is 258 x 258 pixels and 9-bit grey-level data is required which is then to be processed to give a 256 x 256 normalised binary image.

In a typical image processing system this function would be achieved in the following manner. First the analogue signal from the sensor is digitised by a 9-bit A/D convertor to give a grey-level representation of the image. This data is then buffered in a 256 x 256 x 9-bit memory store. Assuming an adaptive threshold technique is to be employed with a patch size of 16 x 16 pixels, a local threshold is then calculated from the stored image for each patch. These are then applied to the image data and the resulting binary image is stored in memory for subsequent processing.

In principle the same function can be implemented without the need for an A/D converter or grey-level image buffer. *Figure 3* shows a simplified schematic of the image normalisation sub-system. This is achieved by reading sequence of 9 image frames during which the 9-bit local thresholds are calculated using a successive approximation technique. As the sensor array is scanned out the selected threshold is fed, via a custom DAC, to one input of a comparator while the analogue image signal is applied to the other. The output of the comparator is a binary representation of the image. The resulting digital pixel stream is analysed to determine the ratio of black and white pixels in each patch using a tally circuit. After each frame each threshold is updated according to the tally result ready for the next frame. This allows a 9-bit threshold to be calculated for each patch, 1-bit per frame, in 9 frames. A final frame is then read and the local thresholds are applied to it resulting in the required normalised binary image.

By suitable sensor addressing and sub-sampling (appropriate to this application) the calculation of the thresholds is achieved within in one frame time. This

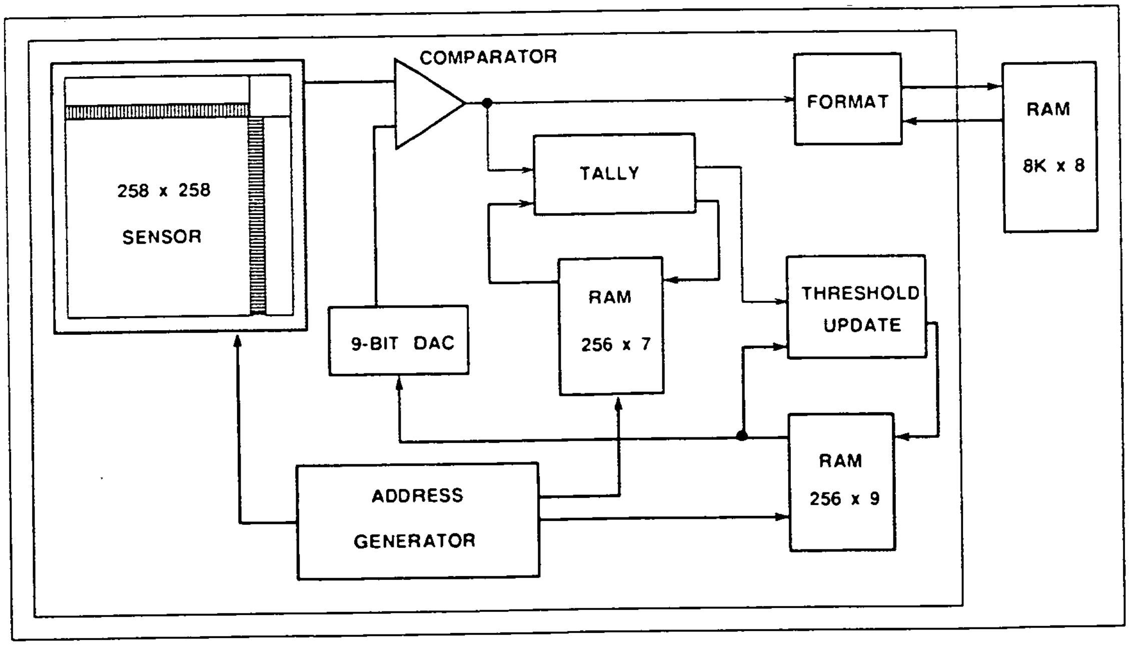


Figure 3: Block diagram of digitisation and normalisation function

tailoring of the algorithm and hardware allows the following savings to be made :- the total RAM requirement reduces from (64K x 9 + 256 x 9 + 64K x 1) = 657,664 bits to (256 x 16 + 64K x 1) = 69,632 bits and the requirement for conversion from a 9-bit flash ADC to a simple DAC and single comparator.

IMPLEMENTATION

The development time for this device was one man year. The device is currently being prepared for fabrication at ES2 (European Silicon Structures) using a 1.2µm CMOS process. The sensor and digital image processing parts of the system have already been independently proven in silicon. The digital logic was implemented using silicon compilation software from ES2 while the full custom elements (sensor, DAC, comparator) were designed by available custom layout design tools. The chip-plot, *Figure 4*, shows clearly the standard cell format of the digital logic and the location of the sensor and memory blocks.

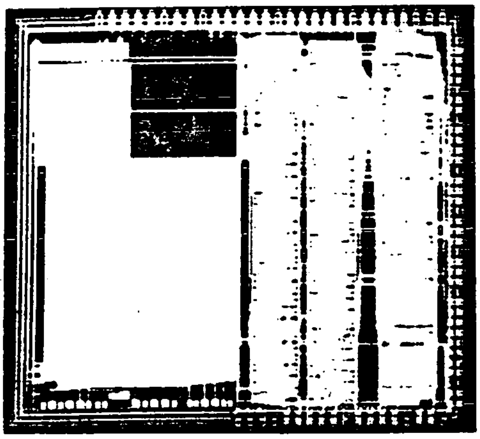


Figure 4: Integrated sensor-processor chip plot

CONCLUSIONS

The ability to integrate the sensor and processing logic as described in this paper facilitates reductions in size , from 525cm^2 to 48cm^2, and component count, from 453 to 23 for the fingerprint matching system. Other features are compared in *Table 2*. This is achieved not just through the usual leverage of integration but also by the additional factor of flexibility of choice of architecture for each component substructure in the system.

ACKNOWLEDGEMENTS

We acknowledge support received from the Science & Engineering Research Council (Grant GR/F 36538 IED2/1/1159). The assistance with the design of the sensor control logic by M. Lu is also acknowledged.

REFERENCES

[1] UK Patent GB 2174831 B, "Skin-pattern recognition method and device", P.B. Denyer, published Dec. 1988.

[2] Renshaw et al., "ASIC VISION," Proc. IEEE Custom Integrated Circuits Conference., 1990, pp. 3038-3041.

[3] Renshaw et al., "ASIC IMAGE SENSORS." Proc. IEEE International Symposium on Circuits and Systems, 1990, pp 7.3.1 - 7.3.4.

[4] Middelhoek and Audet, "SILICON SENSORS", pp. 352 - 355, Academic Press, 1989.

CHIP FEATURES	
Sensor	258 x 258 CMOS Sensor
Functions	3 x 3 unary weighted smoothing filter 9-bit DAC & Comparator Module Adaptive threshold module 20 x 8-bit system registers 8 x 8 correlation cell array Rank-value filter Polling (result analysis module) Microcontroller Interface
Memory	2 x 512 x 16 SRAMs 1 x 16 x 16 SRAM 2 x 704 x 12 ROM
Physical	Package: 160-pin PGA No. Transistors: 272 000 Die size: 11.5mm x 11.5mm Power: 600mW @ 18MHz. Supply: 5v & 0V

Table 1: Integrated sensor-processor features

	ASIC BASED PROTOTYPE	INTEGRATED SYSTEM
No. VLSI / LSI devices	16	3
No. MSI devices	37	0
No. Discrete components	400 (approx.)	20
Power (Ave.)	6W@12MHz.	1W@18MHz.
Area (sub-system)	525 cm^2	48 cm^2
Response Time (typical)	1.5 secs.	0.5 secs

Table 2: Comparison of ASIC prototype and integrated sensor-processor

On-Chip CMOS Sensors for VLSI Imaging Systems

Reprint of a paper presented at VLSI 91

ON-CHIP CMOS SENSORS FOR VLSI IMAGING SYSTEMS

Peter B. Denyer, David Renshaw, Wang Gouyu, Lu Ming Ying, Stuart Anderson

Department of Electrical Engineering, University of Edinburgh,
Mayfield Road, Edinburgh, EH9 3JL, Scotland

Abstract
We present techniques for the integration of vision sensors and processors on common CMOS substrates. This establishes the feasibility of single-chip vision systems which offer reductions in size, power and cost over contemporary techniques.

1. INTRODUCTION

Potential electronic vision applications are widespread; examples include:-

. bar-code and text readers

. security cameras

. image capture for DTP

. biometric verification; fingerprints, faces, etc.

. fax

. production line inspection

. video telephones

. vision subsystems in robots and autonomous vehicles

. consumer camcorders and still cameras

. automotive applications

Virtually all of these applications, and a host of others, are sensitive to cost, size and power consumption. This applies not only to the camera function but also to the subsequent image processing functions, which are invariably computationally demanding.

Commonly such systems are constructed from commodity camera modules, frame grabbers and PC's, or dedicated image processing hardware. These systems can hardly be described as miniature, and their cost and power consumption severely limit their application. Only where production volumes are very large, as in consumer camcorders, do all parameters fall to attractive levels.

Fortunately the advancing capability of VLSI, especially CMOS, technology permits the integration of video-rate A/D conversion and the implementation of powerful custom image processing architectures. Challenging imaging applications, such as fingerprint verification, have become possible within a few tens of cubic inches, consuming a few watts of power. This is still far from ideal and the dominant limiting factor is usually the camera module itself.

It is well known that silicon can act as an excellent photoreceptor over the visible spectrum. The majority of solid-state cameras today use CCD technology (a variant of MOS) which, over two decades, has been highly refined to optimise this function. Some sensor manufacturers use a variant of single-channel MOS technology in which only doping levels are altered to optimise optical performance parameters such as anti-blooming.

Other workers [1,2] have recognised the attraction of implementing sensors in an unmodified CMOS process, permitting the inclusion of the sensor with other control and processing functions on the same chip. Despite encouraging results this technique has never been developed to the point at which the sensor performance matches that of CCD cameras. The purpose of our work is to realise, in unmodified CMOS technology, array image sensors which match the performance of CCD cameras. We have succeeded in this aim and report here the circuit techniques and results that we have achieved, including demonstrator single-chip vision systems.

2. TECHNIQUES

In common with others [3] we use a photodiode sensor comprising an array of MOS transistors, one per pixel, Figure 1. The photodiode is implemented by extending the source region of the transistor. This may be reset and then isolated under control of the MOS transistor gate. All of the gates in each row are driven in common.

Once reset, the reverse-biased (photo)diode converts incident light into a small photocurrent which gradually discharges the photodiode capacitance. The pixel is read by opening the gate, connecting the photodiode to the MOS transistor drain. All of the drain in each column are connected in common and only one row is read at any time.

Commonly, the column lines are gated through an analogue multiplexer to a single external charge sense amplifier. The requirements of this amplifier are daunting considering that high-speed and wide dynamic range must be achieved from a charge packet in the pixel which may be of the order of 1fC. Accordingly we have replaced this scheme by providing charge sense amplifiers at the top of each column. These need not work so quickly, since their activation frequency is equal to the line rate rather than the pixel rate, and they are situated as close as possible to the pixel array. Their sole constraints are the need to achieve a good dynamic range and to be realised within the pixel pitch (of the order of 10-20μm). We use a single-ended differential charge integrator which gives a low impedance 1v analogue representation of the pixel charge with a theoretical dynamic range of 70dB. The read time is approximately 500nsec.

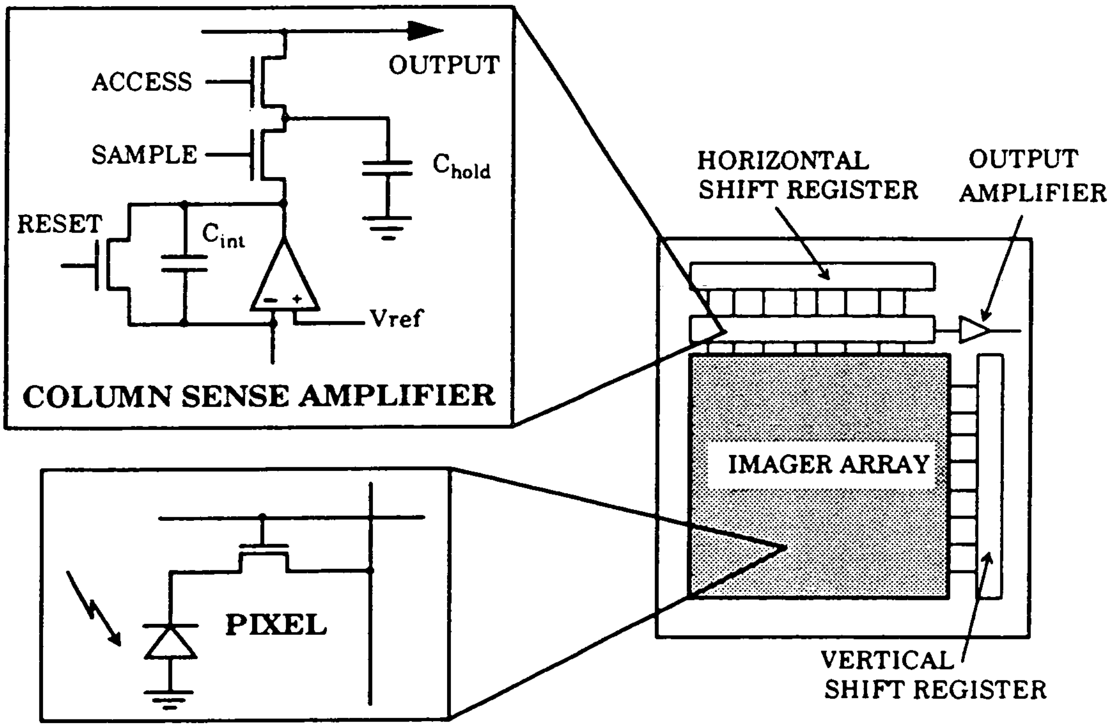


Figure 1. Architecture of a CMOS image sensor with column charge sense amplifiers.

The voltage representation at the output of these sense amplifiers is sampled and stored on a row of capacitors and the information on these is multiplexed out in the conventional manner, with the exception that we implement the output charge integrator on-chip, including a sample-and-hold stage. For applications requiring a composite video waveform it is relatively easy at this stage to include an analogue multiplexer to switch in blanking and sync. levels at appropriate times.

Serially-scanned operation is achieved by adding vertical and horizontal digital shift registers at the periphery of the array and these also must match the pixel pitch. The vertical register successively activates the row lines, whilst the horizontal register controls the sequential pixel read-out within each line. The performance of the array is quite insensitive to these control waveforms and amplitudes, in contrast to CCD, and this is a distinctive advantage of the approach.

Prototype CMOS arrays using this architecture give remarkably good results. They operate over very wide margins of temperature and supply voltage. The single parameter of concern is fixed pattern noise from two sources; threshold variations in the MOS pixel access transistors causing speckles, and mismatches between the column sense amplifiers causing vertical stripes. Without compensation these effects have an rms value around 1% of saturation. In later designs we have eliminated these effects by:-

(i) reducing the applied pixel reset voltage to make the actual reset value independent of the gate potential and gate threshold.

(ii) implementing an offset compensating phase in the common sense amplifiers during idle periods, such as line and frame synchronisation.

These circuit techniques successfully eliminate the fixed pattern noise and overcome a traditional objection to the potential of this approach.

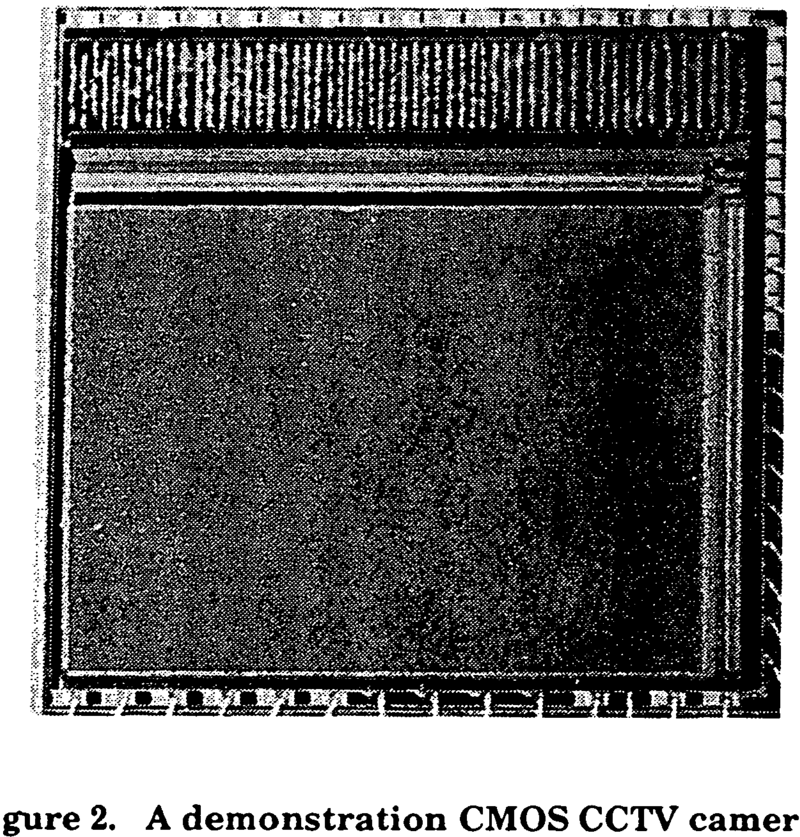
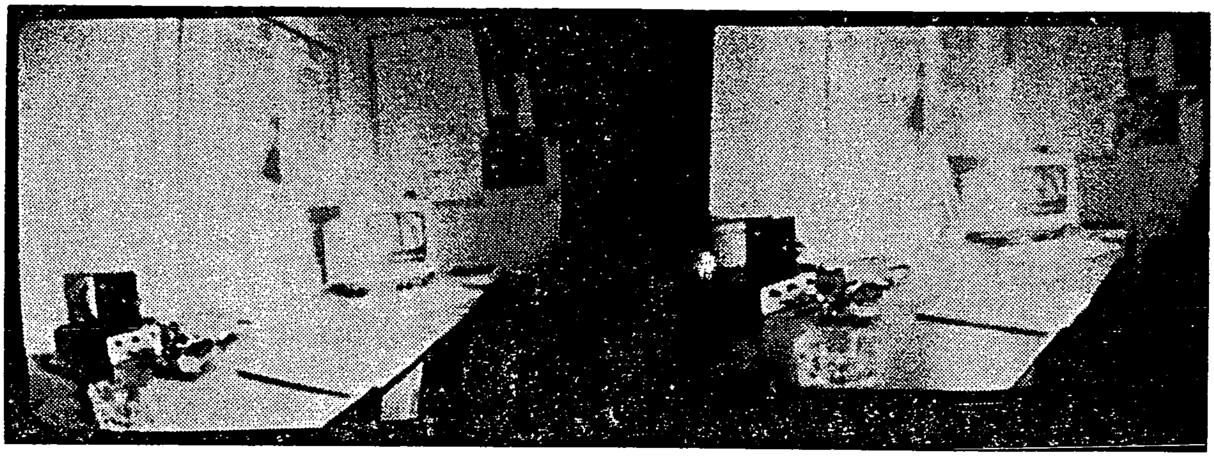


Figure 2. A demonstration CMOS CCTV camera.

Figure 2 shows a completed CCTV camera chip fabricated on a standard 2 micron CMOS ASIC line. The chip contains an array of array of 312 x 287 pixels with timing and automatic exposure control on-chip.



(a) (b)

Figure 3. Comparison of CCD and CMOS camera performance under identical conditions:- (a) CCD (b) CMOS

Figure 3 compares the picture output of this device with an existing CCD camera module. The results are subjectively indistinguishable.

3. AUTOMATIC EXPOSURE CONTROL

By electronically controlling the integration period we can proportionately decrease the sensitivity of the array. We can achieve this through the vertical shift register by controlling the duration, in cycles, of a 'reset' pulse entered at the top of this register. This varies the integration time in steps equal to the line period. We further gate this signal with one of short duration to reduce the exposure time in steps equal to the pixel period, down to a minimum time constrained by the read time of the column sense amplifiers. This is approximately 500nsec, giving a total exposure range of 40,000:1.

If we now alter the exposure time in response to the monitored video output we can implement fully automatic electronic exposure control and avoid the need for mechanical iris control on the lens.

Figure 4 shows a simple scheme for such a control algorithm. The video stream is internally histogrammed in three bins:- very white, average and very black. The exposure is increased or decreased according to whether the image content is judged to be too bright or too dark.

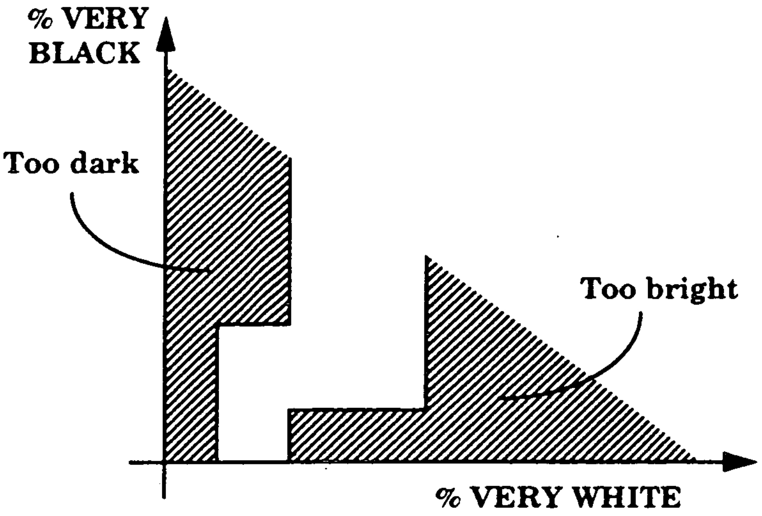


Figure 4. Automatic exposure control decision diagram

We have implemented this scheme (which costs approximately 1000 gates) on several CMOS sensor arrays and obtained satisfactory performance.

4. RESULTS

Over a series of prototypes we have improved the performance of CMOS sensor arrays to match or out-perform typical monochrome CCD performance in most respects. A detailed comparison is given in Table 1. The CMOS data is measured from the CCTV camera demonstrator device shown in Figure 3. The CCD data is compiled from manufacturers data sheets.

	CCD Camera Module (typical)	Integrated Edinburgh CMOS Camera
voltage	12v	5v
power	1W	200mW
minimum pixel size for 1.5μm process	10μm x 10μm	16μm x 16μm
saturation level	20 lux	20 lux
s.n.r.	52dB	51dB
output	composite video 1v p-p	composite video 1v p-p
integration time range	300:1	40,000:1
dark current as fraction of saturation at room temperature, 20msec integration time.	0.005	0.0004
antiblooming factor	100x	100x

Table 1. Comparison of CCD and CMOS sensor performance

5. Application Examples

We report here two examples which illustrate the potential of this technology for integrated vision applications.

Our first example is a low-resolution camera for use in intruder-alarm verification. This device is installed with a Passive Infrared detector and, upon detecting an alarm, transmits a short sequence of video to a control unit which compresses and transmits this data to a remote monitoring station. Within a few seconds of an alarm event a remote observer can deduce its cause and take appropriate action. As greater than 95% of such alarms are false, the provision of video verification will eliminate much unnecessary police action.

Passive alarm units are very low cost items and cost is a primary constraint on this device.

Figure 5 shows the complete intruder-alarm camera module. This includes a 1.5μm CMOS camera chip measuring 5.8mm x 4.0mm combining a 156x100 pixel sensor array with all timing and control electronics on-chip. The device is customised for this application. and this includes drivers which automatically trigger lamp or flash devices to assist in dark conditions. A further novel feature of this device is the use of a miniature glass lens which is bonded directly to the sensor chip surface. This enables a 90 field of view and assists in improving the robustness and cost of this small unit.

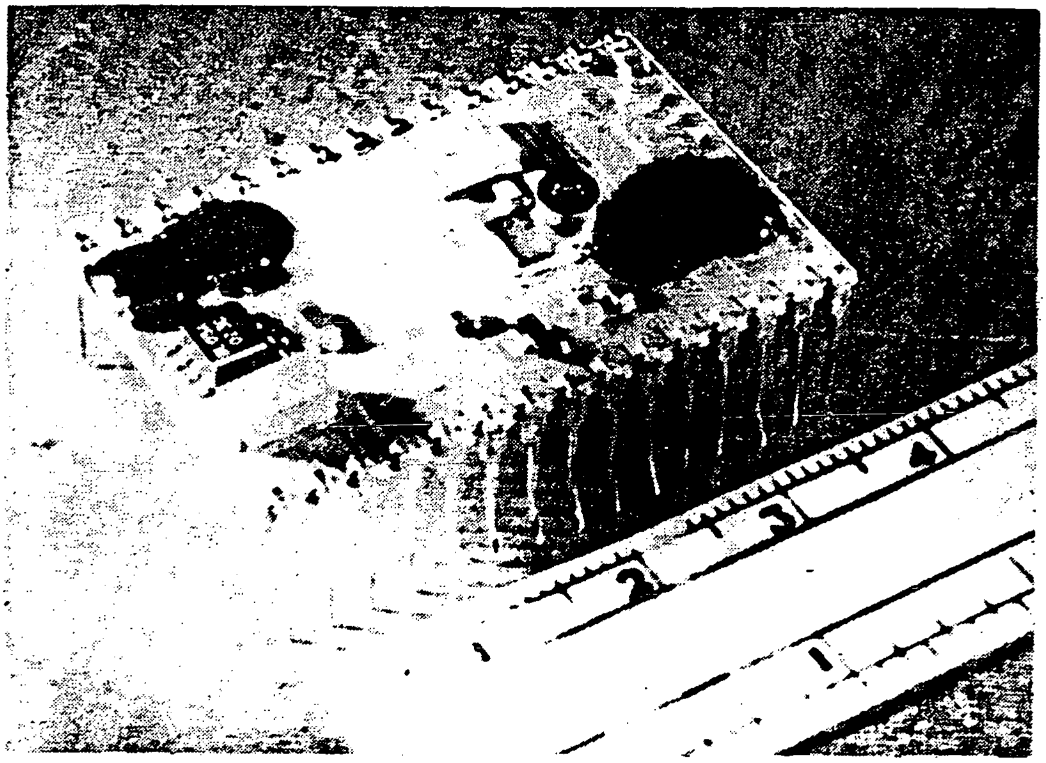


Figure 5. Complete alarm verification camera module using a custom CMOS sensor with a miniature chip-mounted lens

The assembly shown is on a ceramic hybrid substrate and includes a 5v regulator, a clock crystal and approximately one dozen passive components. A simple bipolar stage provides sufficient output impedance to drive 200ft of co-axial cable to the modem unit.

Our second example shown in Figure 6 is a very substantial single-chip vision processing system. This includes:-

. 258 x 258 pixel array.

. Image preprocessing and quantisation to form a normalised binary image.

. 64-cell 2000 Mops/sec correlator array.

. Post-correlation decision hardware.

. 16k bits RAM cache.

. 16k bits ROM look-up table.

With the aid of two external devices (one 64kbit RAM and one 8051 microcontroller) this device performs all of the image sensing and processing functions necessary to capture and verify a fingerprint against a stored reference print within one second.

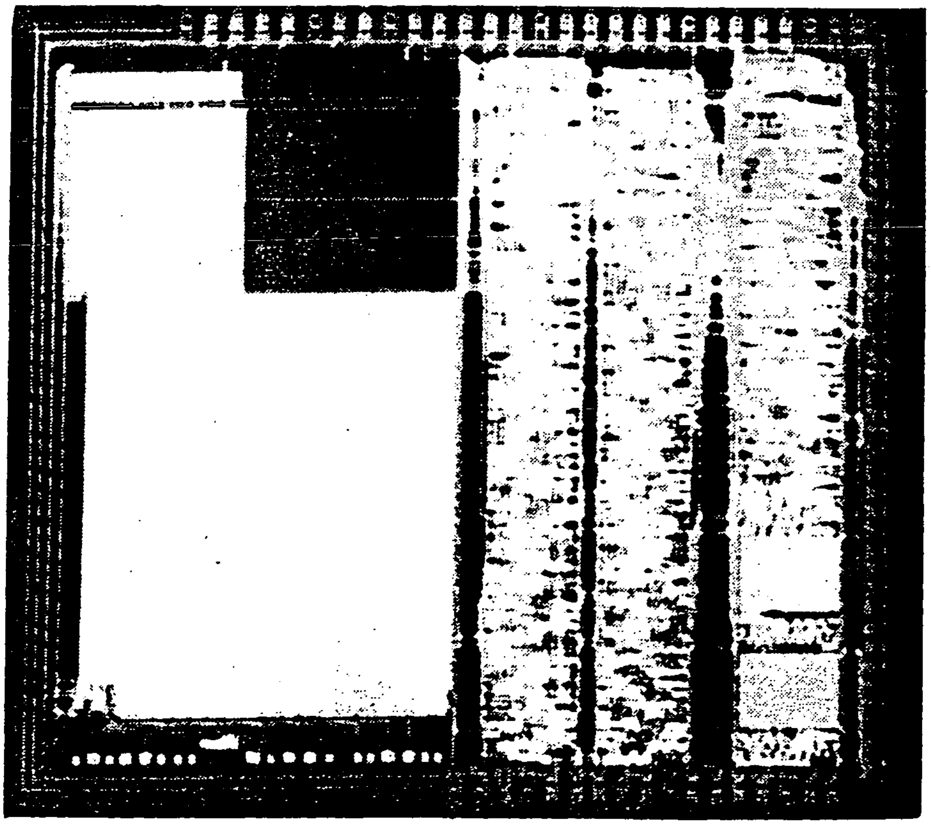


Figure 6. An integrated vision system 258 x 258 sensor array and image processor for fingerprint capture and verification

This is perhaps the best example of our goals; the integration of a sensor and powerful image processor on one substrate.

We have demonstrated several other devices in both 2 micron and 1.5 micron ASIC CMOS technologies and claim that these techniques are portable to any commodity CMOS process.

6. CONCLUSIONS

The aim of 'smart' vision devices, incorporating image sensors and processors in one chip has been substantiated. These integrated devices can be implemented today in unmodified commodity CMOS technology. Vision systems implemented in this way enjoy unprecedented reductions in size, cost and power consumption.

7. ACKNOWLEDGEMENTS

This work was supported by SERC award GR/F 36538, Automated Security Holdings Plc., De La Rue and VLSI Vision Ltd.

8. REFERENCES

1. Lyon, R.F., "The Optical Mouse and an Architectural Methodology for Smart Digital Sensors", VLSI Systems and Computations, 1981, Springer-Verlag, ISBN 3-540-11251-0, pp. 1-19.
2. Tanner, J.E. and Mead, C.,"A Correlating Optical Motion Detector", 1984, Conference on Advanced Research in VLSI, MIT, pp. 57-64.
3. Renshaw, D. and Denyer, P.B., "ASIC Vision", CICC 1990, §7.3.
4. Anderson, S. et al., "A Single Chip Sensor and Image Processor for Fingerprint identification", CICC 1991, §12.1.