DEVELOPMENT OF A CUSTOM MACHINE VISION CAMERA FOR FROTH FLOTATION

by

P J OOSTHUIZEN

June 1999

THESIS PRESENTED IN FULFILMENT OF THE REQUIREMENTS FOR THE MASTERS OF ELECTRONIC ENGINEERING DEGREE AT THE UNIVERSITY OF STELLENBOSCH.

Supervisor: Prof. J.J. Du Plessis

DECLARATION

I, the undersigned, hereby declare that the work contained in this thesis is my original work, and has never been submitted, in part or in its entirety, at any Unversity, for any requirements towards the achievement of any degree.

21/09/1999

P.J. Oosthuizen Date

ABSTRACT

Flotation is one of the most important and versatile mineral processing techniques in the mining industry. Unfortunately, Froth Flotation processes are subjected to a wide variety of process disturbances; some of which are caused by changes in mineral characteristics, and others by variation in operating procedures.

The aim of this thesis is to develop a custom Machine Vision Camera for Froth Flotation plants. Recent research shows that Machine Vision techniques can alleviate many of the difficulties faced by conventional flotation control procedures. The goal of the Machine Vision Camera is to assist in the further development of optimal control strategies for Froth Flotation plants based on the visual features of froth.

As additional assistance in this ongoing process, Discrete Image Cross-Correlation is proposed as a tool to visually extract the two most important parameters of froth. The use of dedicated custom hardware based on FPGA and Transputer technology to implement the Discrete Image Cross-Correlation function for a batch of flotation cells is also presented. Final results showed that a market-standard PC will prove simpler, faster, and more cost-effective than similarly priced dedicated custom hardware.

OPSOMMING

Flotasie is een van die belangrikste en veelsydigste mineraal prosessering tegnieke in die myn bedryf. Ongelukkig word skuim flotasie prosesse aan 'n verskeidenheid proses versteurings onderwerp waar sommige deur veranderings in mineraal eienskappe en ander deur variasies in die bedryf van die flotasie proses veroorsaak word.

Die mikpunt van die tesis is die ontwerp van 'n doelgerigte Masjien Visie Kamera vir skuim flotasie prosesse. Huidige navorsing toon dat Masjien Visie tegnieke baie van die probleme wat met die konvensionele beheer van flotasie ondervind word kan oorkom. Die doel van die Masjien Visie Kamera is om die verdere ontwikkeling van optimale beheerstrategieë wat op die visuele eienskappe van die flotasie skuim gebaseer is te ondersteun.

As addisionele bystand tot hierdie voortdurende poging word Diskrete Beeld Kruiskorrelasie as 'n visuele metode ondersoek om die twee belangrikste eienskappe van flotasie skuim te onttrek. Die gebruik van doelgerigte hardeware gebaseer op Transputer en FPGA tegnologie vir die uitvoering van die Diskrete Beeld Kruiskorrelasie metode vir 'n groep flotasie selle word ook ondersoek. Finale resultate toon dat 'n standaard, kommersiële PR makliker, vinniger en meer koste effektief sal wees as doelgerigte hardeware vir dieselfde prys.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my thanks to Prof. J.J. Du Plessis, for his valuable time, assistance and advice.

Thanks go also to Dr. D. Feng, for his assistance in obtaining practical measurements from a batch flotation process.

I would also like to acknowledge the assistance of Mr. M. Van Olst and Mr. J.J. Eksteen in the clarification of some aspects of flotation process.

Finally, I would like to thank Miss R. Jones for her support throughout.

LIST OF ABBREVIATIONS

1D-DFT	One-Dimensional Discrete Fourier Transform
2D-DFT	Two-Dimensional Discrete Fourier Transform
1D-FFT	One-Dimensional Fast Fourier Transform
2D-FFT	Two-Dimensional Fast Fourier Transform
A/D	Analog-to-Digital
AC	Alternating Current
ADU	A/D Units
AGC	Automatic Gain Control
ASIC	Application-Specific Integrated Circuit
ASSP	Application-Specific Standard Products
B	Blue
CCD	Charge Coupled Device
CDS	Correlated Double Sampling
CKO	Base Clock of the Timing Generator
CL	Secondary Clock of the Timing Generator
CLD	A/D Conversion Clock of the Timing Generator
CMOS	Complementary Metal-Oxide Semiconductor
CPLD	Complex Programmable Logic Device
DC	Direct Current
DFT	Discrete Fourier Transform
DICC	Discrete Image Cross Correlation
DIF	Decimating in Frequency
DIT	Decimating in Time
DR	Dynamic Range
DRAM	Dynamic Random Acces Memory
DSP	Digital Signal Processing
EAB	Embedded Array Block
EDR	Electrical Dynamic Range
FPGA	Field Programmable Gate Array
FT-CCD	Frame Transfer Charge Coupled Device

FFT	Fast Fourier Transform
FFT-CCD	Full Frame Transfer Charge Coupled Device
Gb	Green near Blue
Gr	Green near Red
I/O	Inpout/Output
ID	Line Identification
IPP	Image Processing Platform
ISA	Industry Standard Architecture
IT-CCD	Interline Transfer Charge Coupled Device
JFET	Junction Field-Effect Transistor
LNE	Large Number Emphasis
LPF	Low-pas Filter
MV	Machine Vision
MVC	Machine Vision Camera
MVCI	Machine Vision Camera Interface
MVS	Machine Vision System
NEE	Noise Equivalent Exposure
NGLDM	Neighbouring Grey Level Dependance Matrix Method
NNU	Number Non-Uniformity
ODR	Optical Dynamic Range
PC	Personal Computer
PCS	Primary Control System
PR	Persoonlike Rekenaar
PSIT-CCD	Progressive Scan Interline Transfer Charge Coupled Device
R	Red
RGB	Red, Green and Blue
rms	root mean square
RS-485	Differential communication standard
SEE	Saturation Equivalent Exposure
SM	Second Moment
SNE	Small Number Emphasis
SNR	Signal to Noise Ratio
SRAM	Synchronous Random Access Memory
SUNSAT	Stellenbosch University Satellite

TTL	Transistor-Transistor Logic
TV	Television
VGA	Video Graphics Array
VHDL	Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language
WEN	Write Enable Line
XRS	Synchronisation Clock of the Timing Generator

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Flotation is one of the most important and versatile mineral processing techniques in the mining industry. Originally patented in 1906, flotation has permitted the mining of low-grade and complex ores, which would have otherwise been regarded as uneconomical. Unfortunately, Froth Flotation processes are subjected to a wide variety of process disturbances; some of which are caused by changes in mineral characteristics, and others by variation in operating procedures. As a consequence, Froth Flotation becomes difficult to understand and control.

Recent work in flotation shows that Machine Vision techniques could alleviate many of the current problems encountered in the conventional control of flotation; and that these techniques will be a useful tool in the development of optimal control strategies for Froth Flotation plants.

The primary principle of Machine Vision (MV) is to control a Froth Flotation plant, based on the visual parameters seen in the froth. The aim of this thesis is to develop a custom Machine Vision Camera (MVC) for Froth Flotation plants to help with this further development.

The main focus of this chapter is the introduction of MV as an alternative method for use in the monitoring and control of a Froth Flotation plant, as discussed in current literature. The basic concept of Froth Flotation and how MV can solve many of the current problems experienced with the monitoring and control of Froth Flotation plants will be explained. The most important features of surface froth will also be identified. Some background regarding the principles of Froth Flotation will be given, followed by a discussion of the principle problem areas found in the control of Froth Flotation plants. An explanation of how a Machine Vision System (MVS) can solve many of these problems is also provided.

1.1 Principles of Froth Flotation.

Froth Flotation is a selective process for the separation of valuable solid raw materials from waste (gangue), based on the difference in the surface properties of the raw materials.

Figure 1.1 shows a basic structure for a Froth Flotation cell. Finely ground materials (e.g. valuable minerals or ores) are fed to the flotation cell were they are mixed into the water by an Agitator to form a pulp. The action of the Agitator (Impeller) draws air down into the pulp, where the air stream is sheared into fine bubbles. After the addition of chemical reagents, the differences in the surface properties between the valuable raw materials and the waste within the pulp are made more apparent in a controlled maner. These reagents force the valuable raw materials to become hydrophobic (water repellent) and attach to the air-bubbles in the pulp. The waste is left in the pulp. As the loaded air-bubbles migrate to the surface of the pulp, a stable froth is formed. This is continuously removed to recover the valuable materials.

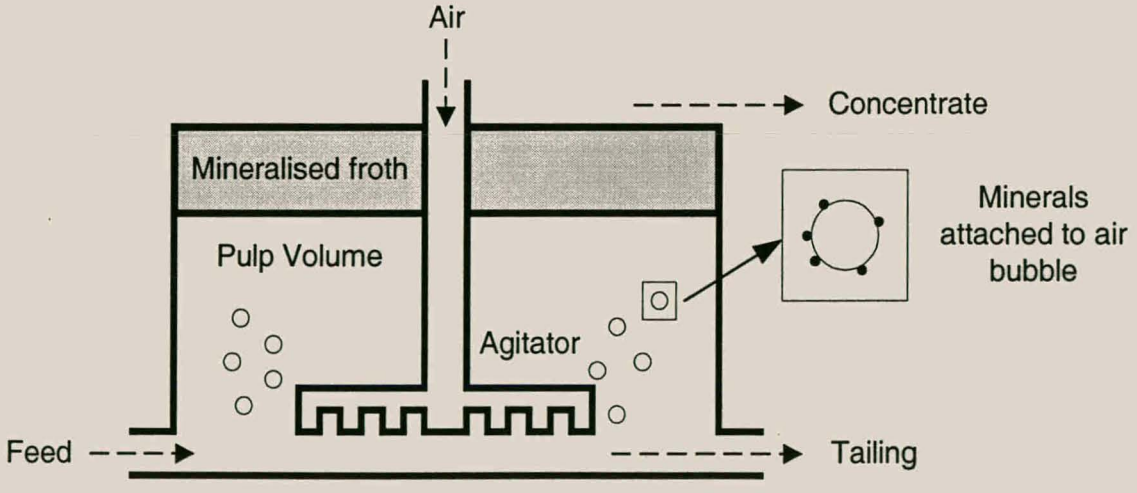

Figure 1.1 Basic structure of a Froth flotation cell.

The key to the successful operation of a flotation process lies in the correct selection of the chemical reagents (*collectors*, *frothers* and *regulators*) to be added to the pulp; since these chemicals control the processes involved to create a stable froth with a high mineral content.

Most minerals are not hydrophobic in their natural state, thus flotation reagents must be added to the pulp. The most important of these reagent are the *collectors*, which adsorb on mineral surfaces, rendering them hydrophobic. In this way they force the attachment of minerals to the air-bubble surface. After the addition of a *collector*, the stability of the air-bubble mineral attachment, especially at the pulp surface, depends to a considerable extent on the efficiency of the *frother*. If the stability of the surface froth is too low the air-bubbles will burst, dropping their valuable content back into the pulp before the minerals can be recovered. On the other hand *regulators*, also known as *modifiers*, are used extensively to modify the action of the *collector* by making the *collector* more selective towards certain minerals. This is achieved either by intensifying or by reducing the hydrophobic effect of the *collector* on the mineral surface.

In practise, an industrial Froth Flotation plant consists of groups of Froth Flotation cells arranged in a series of banks as shown in Figure 1.2. Figure 1.3 shows a photo of an actual bank of flotation cells.

Pulp is fed to the first cell of the bank where some of the valuable minerals are removed as froth. The overflow from this cell then progresses to the second cell where more minerals are removed. This process continues down the bank of cells until the mineral content of the pulp is depleted and overflows the last cell of the bank.

As the pulp progresses from cell to cell the pulp level is increased to keep the froth level constant thoughout the bank, since the froth depth decreases as minerals are removed from the pulp.

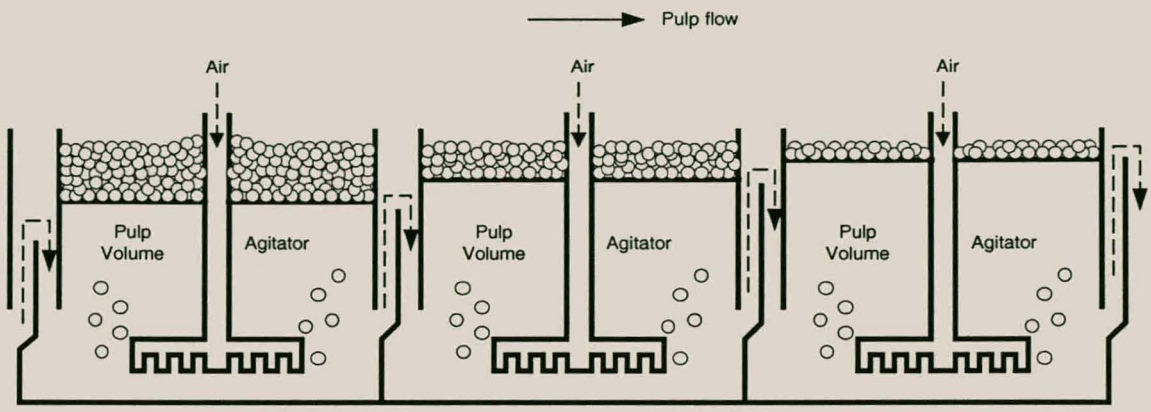

Figure 1.2 Simple diagram of a bank of Froth Flotation cells.

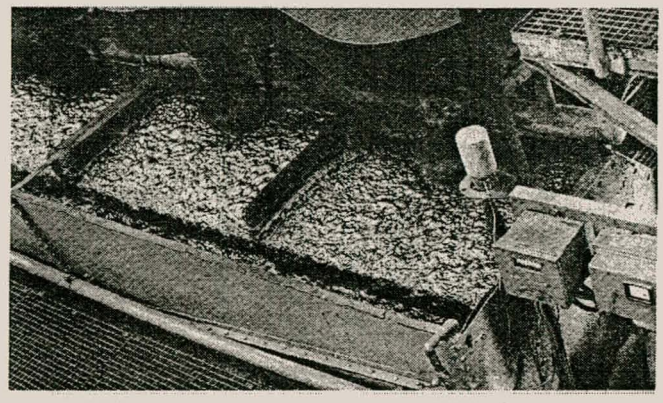

Figure 1.3 Bank of flotation cells.

Figure 1.4 shows a diagram of the most simple flotation circuit for a flotation plant. Feed (finely grinded ore or minerals) is fed to the bank of cells. The froth heights of the first few cells are kept high to produce a high-grade concentrate, since there will be enough hydrophobic minerals to sustain such froth. As the pulp is progressively depleted of floatable minerals, the pulp level is raised from cell to cell. The last few cells (scavengers) in the bank will contain relatively low-grade (low mineral content) froths, consisting of weakly hydrophobic minerals. These cells are often recirculated to the head of the system to ensure the maximum recovery from the bank of cells.

This type of circuit is typically only used when the waste is relatively unfloatable, and requires extremely careful control to produce an even grade of concentrate for any variation in the head grade.

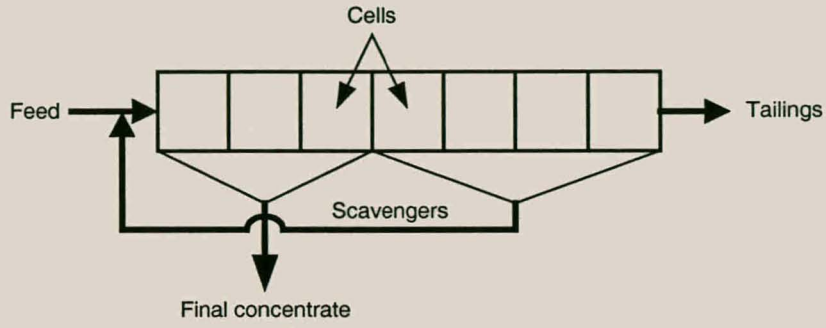

Figure 1.4 Flow sheet of a simple flotation circuit.

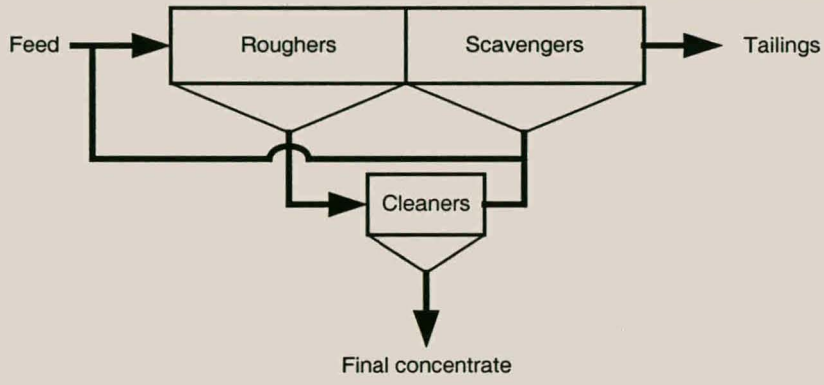

Figure 1.5 Diagram of a rougher-scavenger-cleaner system.

The rougher-scavenger-cleaner system is preferred to the circuit portrayed in Figure 1.4. This system is shown diagrammatically in Figure 1.5. The concentrates of the roughers are first diluted and then refloated in the cleaner cells where the froth depth is kept high in order to produce a high-grade concentrate.

This circuit is typically used when the waste has a tendency to float and is difficult to separate from the minerals. The circuit is also useful to float ores that need a high amount of aeration at the end of the bank to ensure a profitable recovery. Wills (1981) provides a more detailed discussion of the principles of flotation and flotation circuits.

1.2 The Need for Machine Vision in Industrial Froth Flotation.

The following section will discuss the current problems experienced in industrial Froth Flotation control and will explain why MV can help to solve many of these problems. The discussion is based on the work done by Moolman et al., (1995a, 1995b, 1996).

The aim of Moolman's research is to explain and clarify the potential for the use of a MVS in an industrial Froth Flotation plant. This potential is explained through the use of a textural description of the surface froth; and the classification of froths based on features extracted from images of surface froth using neural networks techniques (Moolman 1995a).

Moolman et al., (1996) emphasise the potential of a MVS by giving an indication of the problem areas in the conventional control of Froth Flotation plants, followed by the reasons why a MVS should be able to solve these problems. These problem areas include:

- Unpredictability of flotation circuits.

  Flotation processes are well known for their high susceptibility to process disturbances. These disturbances are often caused by changes in the mineral characteristics and by variations in the operating procedures of the flotation process. Changes in the flow rate, densities, size distributions and shapes of particles, surface properties, and the compositions of the raw materials will typically have a significant effect on the plant performance. Additionally, errors caused by actuators, inaccurate measurements, and the malfunctioning of equipment are also well known sources of process disturbances.

- Lack of suitable instrumentation.

  A major problem often encountered is that the instruments used are not always available, reliable or accurate. Some instruments (pressure sensors, conductivity probes, and ultrasound devices etc.) are contacting devices and are therefore easily affected by variations in the physical properties of the pulp in the froth phase. In addition, these instruments also require a high degree of maintenance (cleaning, unclogging etc.) to keep them functioning properly. Other sensors suffer from long delay times between the actual measurement and the result.

- Complexity of control strategies and control devices.

  Moolman et al., (1996:136) report that from experience it is known that most monitoring systems in industrial flotation plants develop technical difficulties on a regular basis, which are often not solved. This results in the system becoming obsolete. Subsequently, the robust control methods used for flotation that deliver a suitable level of performance still require a large amount of further development; while the model-based control methods are complicated, and require skilled control engineers to implement and maintain.

- Problem areas with the conventional Froth Flotation control.

  The most common control strategies include frother and depressant control, collector control, and pulp level control. According to Moolman et al. (1996:138), the primary disadvantage of the frother and depressant control is their sensitivity to process disturbances. Even disturbances of short duration will have a severe effect on the performance of the flotation circuit. Similarly, collector control is sensitive to variations in the mineral or ore feed of the process. For both methods, the problem is attributed to the long delay induced by the conventional measurement systems between the measurement of the process variables, and the actual adjustment of the control variable. On the other hand, the pulp level control suffers from the disadvantage that most of the devices used to measure the pulp level are contacting devices, and are therefore sensitive to variations of the pulp phase.

- Inadequate measurement of process set points.

  A typical control cycle for a flotation plant consists of sets of decisions regarding a set of measurement points, a waiting period for the transient dynamics of the process to settle, a period of measurement and evaluation, and the estimation of new suitable measurement points. Moolman et al., (1996:137) report that it is often found that plant operators make the periods for settling and measurement too small; resulting in sub-optimal control of the flotation process. In practice, not all points of a process can be measured, and most on-line monitoring systems only measure the assays of the concentrate collected from a whole bank of flotation cells. This implies that a significant time delay is added to the process measurements and that process deviations of a single flotation cell will not always be detected.

Moolman et al., (1996) give a more detailed discussion of the problem areas and control strategies of froth flotation. Based on the above mentioned problem areas, Moolman et al., (1996) suggest that a simple, reliable control system, which is easily understood by flotation process operators, can be a solution. The concept of Moolman's control strategy is to integrate the measurement and control procedures "within the context of visual characterization (sic)" of the surface froth (Moolman et al., 1996:139).

To further develop this concept, Moolman et al., (1995b) demonstrate through the analysis of images of batch flotation processes and flotation plants, that a relationship exists between the visual appearance of surface froth and the performance of a flotation plant. The visual appearances used were the colour, bubble size, texture, stability and mobility of the surface froth.

The significance of these parameters can easily be described on a qualitative level:

- Colour

  From the colour of the froth it is possible to obtain information about the mineral species and their concentrations in the surface froth. The concept of using froth colour can easily be understood in the sense that certain minerals have specific colours, and changes in the concentrations of these minerals in the froth will cause changes in the

in the froth colour. Moolman et al., (1995b:3508) report that the colour is only an important parameter in MV if the relationship between the specific mineral content and froth colour is clearly understood.

- Bubble size

  The bubble size distribution in the pulp is important, since it affects the probability of adhesion between the minerals and the surface of the bubbles. Two assumptions made in this regard are that relationships between processes occurring inside the pulp and the changes on the froth surface, and between the size of the bubbles in the pulp and the surface froth do exist (Moolman et. al., 1996:140). Qualitatively, these assumptions make sense, since processes inside the froth will eventually cause changes in the surface.

- Stability and Mobility

  The stability of the surface froth is one of the most important factors in the final separation of minerals from the pulp (Moolman et al., 1996:146). If the froth is too unstable, the high burst rate of bubbles on the surface of the froth will cause the bubbles to drop their mineral content back into the pulp before the minerals can be removed from the froth. Conversely, a highly stable froth will cause excessive build up of large froth bubbles and a high volume for the surface froth, causing the slow recovery of minerals. On the other hand, the mobility of the surface froth is responsible for the unwanted recovery of waste (Moolman et al., 1996:145). This is because the higher the mobility of the froth, the less time there will be to allow the waste materials to sink back into the pulp before the froth is removed. Intuitively, it can be argued that a relationship between the mobility and stability of froth exists; since typically the lower the mobility of the froth, the higher its stability.

- Texture

  With the above parameters in mind, it is clear that the texture of the surface froth will be altered with changes in the froth structure and can therefore be an indicator of changes in process parameters.

Moolman et al., (1995b, 1996) give a more detailed technical discussion of the importance of these parameters and their relation to the parameters of a flotation process. Further analysis by Moolman et al., (1995b:3507-3511) show that the mobility and stability, followed by the average bubble size are the most important of the above mentioned froth features. Based on these results, Moolman et al., (1996:139) propose a MVS that monitors each individual cell of a flotation circuit. This will allow the early detection of process deviation in an individual flotation cell. The MVS will be a non-contacting device using a simple, reliable sensor (TV or CCD camera), which captures images of the flotation circuit and extracts the relevant features from the froth. As a result, a MVS will enable the fast feedback of process parameters to the flotation control system and therefore eliminate the long delays caused by conventional measurement systems.

Other advantages of a MVS, (Moolman et al., 1996:139) are that the results of off-line analysis can be combined with the results of previously captured images and extracted features to help with the more efficient interpretation of froth images and the training of plant operators. A MVS will also help to develop new research insight in flotation, since characteristics previously determined by human judgement can now be described formally in terms of computed feature parameters.

1.3 Conclusion.

Froth Flotation is an important and valuable mineral processing technique for the recovery of low-grade and complex ores.

In practice, it is found that various problem areas exist which make the efficient control of flotation circuits a difficult task. Unpredictable behavior of flotation circuits; lack of suitable instrumentation; complexity of control strategies and control devices; and inadequate measurement of process parameters, are all contributory factors.

After a literature study regarding the possible control of flotation based on the visual interpretation of the surface froth through the use of a MVS was undertaken; it was concluded that a MVS will be able to solve many of the current problems. This is due to the fact that a MVS is a non-contacting device requiring a low level of maintenance; allowing the quick feedback of process parameters to the flotation control circuit. From this study it was also realised that of the features used (colour, bubble size, texture, stability and mobility) to visually characterise froth, the two most important features are the stability and mobility of the froth.

The remainder of the thesis is structured as follows:

- Chapter 2 evaluates the desirable sensor characteristics of a custom MVC to ensure the successful monitoring of Froth Flotation. Firstly, a selection criterion for choosing a sensor is established. The specific choice of sensor and a discussion of its characteristics follow.

- The focus of Chapter 3 is the detailed implementation of the MVC, based on the design goals and methodology used. A simple implementation of a custom Personal Computer interface for the MVC is also introduced.

- Discussion in Chapter 4 centres around the evaluation of the imaging capability of the MVC, and determines whether the camera will be able to suitably monitor surface froth to ensure the successful extraction of the most important froth features.

- Chapter 5 introduces the concept of Discrete Image Cross-Correlation as an additional method for the extraction of the two most important features of froth. Practical images of a batch Froth Flotation cell, and video footage of an industrial Froth Flotation cell are used to evaluate this method.

- The possibility of developing a custom Image Processing Platform, using Transputer and FPGA technology, for the custom MVC is considered in Chapter 6. The function of the platform is to implement the Discrete Image Cross-Correlation method on dedicated hardware.

- Chapter 7 concludes the thesis with a summary of all the results. Recommendations for the future development of the MVC are also offered.

CHAPTER 2

DESIRABLE SENSOR CHARACTERISTICS FOR A MACHINE VISION CAMERA FOR FROTH FLOTATION.

In this chapter, an appropriate CCD sensor is chosen for use in the development of a custom MVC for Froth Flotation. The choice of CCD sensor will ensure that the MVC will capture suitable images of surface froth allowing for successful extraction of the most relevant features of the froth as discussed in Chapter 1.

The criteria used to select an appropriate CCD sensor are explained, followed by a brief discussion regarding the type of image processing undertaken in Froth Flotation. The requirements placed on the choice of CCD sensor by the selection criteria are outlined, after which the specific choice of sensor and its characteristics are discussed.

2.1 Selecting a CCD sensor for Froth Flotation.

There are various types of solid-state CCD sensors available in the current imaging industry. However, not all are suitable for MV in flotation. It is therefore important to have a well-defined set of selection criteria before a decision regarding the sensor is made.

Two of the most important questions that should be asked before determining the requirements of a CCD sensor are:

- What form of image processing is to be implemented; and secondly
- What information does the image-processing algorithm require?

Once the aforementioned questions have been answered, the specific requirements of a CCD sensor can be addressed. These are as follows:

- Spectral response
- Image sample rate
- Sensitivity
- Dynamic range
- Resolution
- Sensor type

2.2 Image Processing in Froth Flotation.

Image representation is concerned with the characterisation of the amount of information represented by each pixel. An important consideration in this respect is the fidelity or intelligibility criteria for measuring the quality of an image or the performance of a processing technique. Specifications of such measures require models for perception of contrast, spatial frequencies, colour etc. Only once the fidelity criteria have been determined can a vision system be properly designed, since it gives an indication of the variables that should be measured most accurately.

In general, two types of image modeling structures are available through which an image can be represented: local models, and global models. With respect to global modeling, the image is considered as a composition of several objects, where various objects are detected in the image. The model itself gives the rules for defining the relationship among the various objects. On the other hand, local modeling is a classical method of signal representation, and is generally sub-divided into deterministic and statistical models (Jain, 1988). Jain (1988:5) reports that a statistical model is often more suitable for most natural (random) textures, since it describes an image as a member of a group, often based on some statistical property of that image. This permits the development of algorithms that are useful for an entire class or group of images rather than for a single image. An important fact worth noting is that statistical methods place low constraints on the quality of an image; because algorithms based on these methods are based on the

overall structure and spatial relationships of the image, not on the individual component structure thereof. An effective use of statistical models in image processing is to consider the images to be spatially varying or piecewise spatially invariant (Jain, 1988).

In the case of Froth Flotation as mentioned in Chapter 1, the surface froth of a flotation cell can be characterised by its texture. Since in practice, plant operators often "base crucial control decisions on the visual appearance of the (surface) froth"; a textural approach is often followed to extract the relevant froth features (Moolman, 1995a:75). From Chapter 1 it can be seen that Moolman et al. find this to be an effective way to characterise the local properties of surface froth. This is done by distinguishing various spatial relationships between the neighbouring gray pixels of an image, based on the histogram features of the surface froth (Moolman et al., 1995b).

Since the basic class of imaging algorithms implemented in Froth Flotation has now been identified, the specific properties of the imaging sensor can be addressed in further detail.

2.3 Spectral Response.

In every MVS, it is important to determine the type of information content needed and the ways in which this information will be used. The fact that the principle of image analysis in flotation is based on the visual perception of the surface froth as a texture seen by the plant operators was previously discussed. In Chapter 1, it was indicated that flotation plant performance can be related to the visual appearance of the surface froth. The fact that monochromatic and RGB colour content of images can sometimes be related to the grade (quality of mineralization) of the froth was also mentioned.

However, a further important consideration to keep in mind is that a flotation MVS can be used to show inexperienced plant operators how the optimal froth structures should appear. A MVC may not be useful to train a plant operator if the camera detects only features that are not visible to the operator. With the information and objectives outlined above in mind, it is clear that it will be of significant advantage if the MVC is able to

provide both monochromatic and RGB colour information within the visual spectrum (400nm – 700nm).

Although it is possible to obtain a monochromatic image from a colour CCD; a monochromatic CCD is typically used in industrial MV applications, because it has an improved capability feature to detect spatial variations. The reason for this is that monochromatic sensors have higher resolutions; better signal to noise ratio; increased light sensitivity; and greater contrast than similarly priced colour sensors (Edmund Scientific, 1998a).

Most commercial and scientific CCD sensors have a suitable spectral response in the visual spectrum. This parameter is therefore not a primary factor for consideration when choosing a CCD sensor.

2.4 Image Sample Rate (Frame rate).

The sample- or frame rate at which a CCD sensor will be required to operate is dependent on the dynamic response of the scene being imaged. In the case of Froth Flotation, the important dynamic response is the natural dynamics of the surface froth of a flotation cell. Unfortunately, the dynamic response of the surface froth is strongly dependent on the characteristics of the minerals (mineral type, size etc.) being separated; the control applied (addition of chemical reagents etc) to the flotation process; and the type of flotation cell being used (size, method of air addition etc.).

Two important dynamic components of the surface froth are the translation speed and the bubble burst rate. Typical values for the bubble burst rate can range from a static value to several Hertz, while the translation speed of the surface froth can range from static to several centimeters per second. To ensure that the bursting of bubbles and the translation of the surface froth does not cause visual distortion effects in an image (aliasing, image smear etc.) a relatively high frame rate is required. A high frame rate will also make the MVC versatile for a wide range of Froth Flotation processes.

Since most commercial and scientific CCD sensors are capable of frame rates of 25 frames/s or higher, the frame rate is not a primary parameter to consider in selecting a CCD sensor for imaging Froth Flotation.

2.5 Sensitivity.

The sensitivity, also known as the photometric responsitivity of a CCD sensor, gives an indication of the ability of the sensor to convert incident light (photons) into voltage. For a fixed sensitivity value, the amount of charge accumulated per pixel is directly proportional to the integration period of the CCD sensor (shutter speed). The higher the shutter speed; the lower the amount of charge will be that is accumulated per pixel.

In most Froth Flotation plants, additional illumination will be required, due to the fact that there is not always a good source of light for imaging. By providing this, features of the surface froth can be visually enhanced, which helps with the extraction of the features from images of the surface froth (Edmund Scientific, 1998b). Moolman (1995a:87) finds this to be the case when monitoring a flotation plant, placing a camera and spotlight perpendicular to the surface of the froth. The additional illumination helps to minimise problems found with bright light reflections and shadows, while increasing the contrast of the imaged surface froth.

Since additional illumination in MV of Froth Flotation will probably be used; the sensitivity of CCD sensor is not a crucial element in the selection of a sensor, and can be compensated for by additional external illumination. The sensitivity of the CCD sensor can also be enhanced by making use of additional signal processing (by increasing the gain and lowering the noise of the CCD sensor) when designing a MVC.

2.6 Dynamic Range.

The dynamic range is a very important parameter of a CCD, and defines the maximum Signal to Noise Ratio (SNR) of the sensor (image quality). The dynamic range also gives an indication of the maximum amount of digitisation levels that can be used for a given sensor. This is an important fact to keep in mind due to the possibility that the rest of the camera electronics can be selected or designed to have a higher SNR than that of the CCD.

Typically, the choice of the minimum amount of digitisation levels for a MVC is not primarily based on the dynamic range of the CCD sensor, but on the image processing system and the requirements of the image processing algorithms. The higher the amount of digitisation levels, the slower the image processing system's throughput.

Paragraph 2.2 discussed the typical type of image processing done in Froth Flotation, and pointed out that textural based methods are commonly used to extract features from the surface froth. The speed and efficiency of these image processing methods are highly dependant on the resolution (amount of pixels) and dynamic range (bits per pixel) of the image. The higher the dynamic range, the more processing is required. Using a non-standard bus width can also cause problems when interfacing a MVC with an image processing system, since most common processors work with 8-, 16- and 32-bit bus architectures.

Based on the above reasoning, it was decided that 8-bit digitisation should be used, since it is a standard bus width, and can easily be interfaced with many image-processing systems (PC's or custom hardware). This bus width is also a common standard among custom MVCs and delivers a high image quality.

The choice of 8-bit digitisation implies that a dynamic range of minimum 48dB to maximum of 60dB is required from the CCD sensor[1]. Fortunately, this places a low constraint on the choice of sensor, since even most commercial CCD sensors can achieve this range.

2.7 Resolution.

The choice of the required spatial resolution of the CCD sensor can be based on two factors. First of these is the physical properties of the monitored scene; which entails the smallest resolvable feature that has to be detected, the area of the scene, and the distance from which the scene is imaged. The second factor is that the lower the resolution of the sensor, the less data will have to be processed by the image processing system. This will increase the throughput of the image processing systems.

The fact that the typical type of image algorithms implemented in flotation is based on the overall statistical properties from images of the surface froth, and not the properties of any individual object of the surface froth was explained in paragraph 2.2. Furthermore, paragraph 1.2 gave an indication that the bubble size is a useful secondary indicator of processes occurring within the froth.

Typical bubble sizes to be expected in industrial flotation range from 10mm – 30mm when efficient flotation is achieved; and can be larger than 50mm if the froth become excessively stable (Moolman, 1995a:31).

However, the question may be asked whether it is not necessary to extract the size of every bubble to determine the average bubble size? Moolman (1995a:75) reports "It is highly unlikely that experienced flotation plant operators attempt to visually estimate the froth bubble size distribution in any particular cell. It is more likely that they classify froths into one of a number of categories ranging from "very large" to "very small" on a comparative basis". Moolman (1995a:89-94) continues to show that it is possible to

[1] The calculation of these dynamic range values can be found in Appendix A.

extract the average bubble size from images of surface froth by using the overall spatial characteristics of the surface froth, without the need to individually extract every bubble's size.

The choices of the area imaged in Froth Flotation are primarily based on physical and practical constraints of a particular flotation cell. The surface of the froth being imaged must convey only the natural dynamic response of the froth, and not the response caused by any form of external agitation (addition of air, mixing of froth pulp and removal of the surface froth). For these reasons, the typical size of areas imaged in flotation cells can range from 0.2m x 0.2m to 0.3m x 0.3m from distances ranging between 1m to 2m. The specific choice of imaging distance is to ensure that spray from the froth does not influence the MVC, while keeping the resolution requirement of the CCD sensor to a minimum.

Since the smallest important feature of the surface froth is the bubble size, most commercial and scientific CCD sensors, with the right choice of optics, can approximately resolve 8 pixels (for a colour sensor) or 16 pixels (for a monochromatic sensor) over a 10mm bubble when imaging an area of 0.3m x 0.3m from a 2m distance.

To estimate the relative efficiency of this resolution, it can be compared with the Johnson Criteria (RCA, 1974:120). The Johnson Criteria state that a system must resolve 2 pixels of the CCD sensor across the critical dimension of an object in order to detect the object; 6 pixels in order to classify the object; and 14 pixels in order to identify the object. In the case of froth bubbles, the critical dimension is the bubble size, and therefore it should not be a problem to classify or even identify a bubble uniquely with a CCD sensor. Further discussion regarding the resolution of the MVC will be undertaken in Chapter 4.

2.8 Sensor Type.

Since the general characteristics of a CCD sensor and their role in imaging Froth Flotation have now been discussed, the specific types of sensors and their use in monitoring flotation can be addressed. The five sensors under discussion are:

- Photodiode image sensor
- Frame transfer CCD sensor
- Full Frame transfer CCD sensor
- Interline transfer CCD sensor
- Progressive Scan Interline transfer CCD sensor

2.8.1 Photodiode Image Sensor.

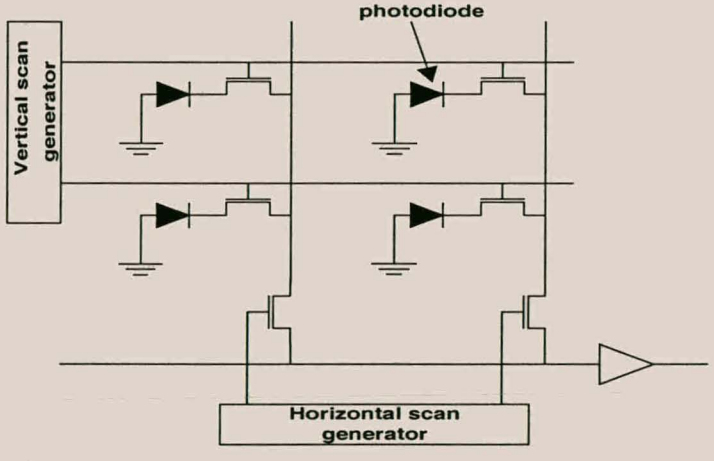

Figure 2.1 Photodiode image sensor.

Photodiode image sensors consist of a two-dimensional array of pixels, each of which is equipped with a photodiode and transistor. The transistor works as a switch, controlled by a digital addressing circuit. Pixels are connected through their transistor to a common line, leading to the amplifier where the charge from the selected photodiode is converted to a voltage. The advantage of this type of sensor is that it has a moderate fill factor

(sensitivity) and the pixels of the sensor can be accessed randomly. The disadvantages of the sensor are that shuttering cannot be implemented; images (pixels) must be clocked out at a higher rate than CCD sensors to image the same dynamic scene with minimum reduction in image quality. Additionally, a photodiode sensor also has a lower dynamic range than CCD sensors. Photodiode image sensors are therefore typically used only in special applications where random access to pixels is required (ASPRS, 1994).

2.8.2 Frame Transfer CCD (FT-CCD).

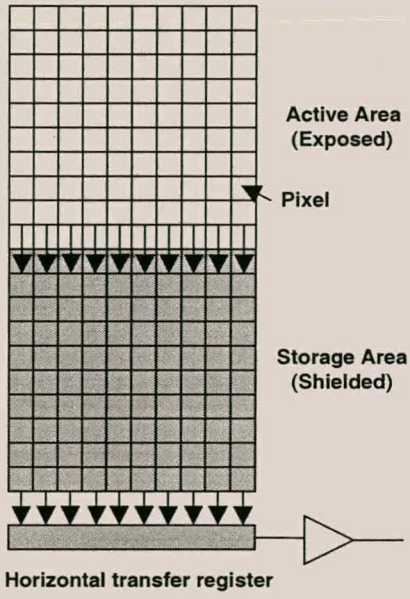

Figure 2.2 Frame Transfer CCD.

The FT-CCD is a simple two-dimensional array of pixels, with a dedicated light shielded storage area separated from the imaging area. As light falls on the image area of the sensor, charge is accumulated. After an integration period, the charge from the image area is transferred line after line to a storage area. The charge from the storage area is then shifted to a shielded horizontal transfer register, which transfers the charge packets (pixels) serially to the output amplifier. The amplifier converts the charge from the horizontal transfer register to a voltage.

The double image area of the FT-CCD gives the sensor the ability to capture an image, while a previous image is clocked out. The double image area also has the added advantages of maximising the pixel fill-factor (sensitivity) of the active imaging area and enabling the sensor to have limited electronic shutter control. A disadvantage of the FT-CCD is that while an image is transferred from the active image area to the storage area, charge is still captured. This degrades the image quality.

The typical tradeoffs of the FT-CCD architecture are high dynamic ranges (> 60 dB); high sensitivity and frame rates (100's of frames/s) compared to increased die size and corresponding sensor and camera cost (DALSA, 1997a). The FT-CCDs are not widely used in the commercial TV and video market and are generally used for medical, scientific and MV applications requiring high image quality.

2.8.3 Full Frame Transfer CCD (FFT-CCD).

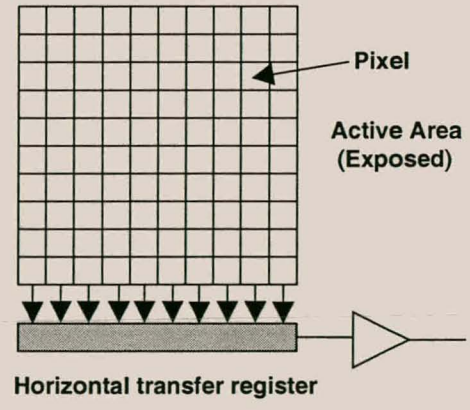

Figure 2.3 Full Frame Transfer CCD.

A FFT-CCD functions similarly to a FT-CCD, except that the imaging area also functions as the storage area. The advantage of this architecture is a very high pixel fill-factor (sensitivity) and small die size, resulting in lower sensor and camera costs. The major disadvantage is the absence of a light shielded image storage area, making electronic shuttering within the sensor generally impossible. Shuttering must be performed with a

strobe or external shutter mechanism, making the sensor complicated to use. FFT-CCD sensors must also be clocked at higher rates than FT-CCD and IT-CCD sensors to image the same dynamic scene, with minimum loss of image quality. The FFT-CCD sensor is typically used for similar applications as the FT-CCD sensor.

2.8.4 Interline Transfer CCD (IT-CCD).

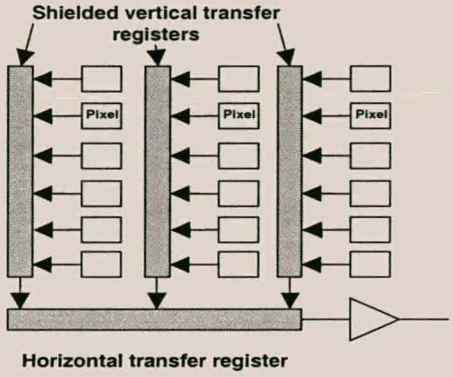

Figure 2.4 Interline Transfer CCD.

IT-CCD sensors also have two-dimensional pixel array structure, where each pixel is made up of an optically sensitive region and a light-shielded storage region. The vertical transfer of charges to the vertical transfer registers, and the horizontal transfer of charges to the amplifier occur shielded from light. The image is therefore not corrupted during photocharge transfers.

These sensors have the advantage of the fastest electronic shuttering performance with minimal increase in sensor die size; low sensor and camera costs, with tradeoffs in fill-factor (sensitivity) and dynamic range (DALSA, 1997a). The major disadvantage of IT-CCD is that the sensor requires two shutter events (two image fields) to capture an image. This effectively halves the sensor's vertical resolution and capability to image dynamic scenes. In dynamic image capture, by the time the second field of the image is scanned or output, the scene or object has moved. This causes a blurring effect (smear) when the two fields are combined (interlaced) to create the full image. Interlaced images of even static objects can introduce noticeable "jitter". An additional disadvantage of IT-CCD sensors

is their fixed TV resolutions, since they are primarily used for TV and video applications. Colour IT-CCD sensors also require additional signal processing to extract the individual colour content of a pixel. Most commercial CCD applications and standard RS170/NTSC and CCIR/PAL cameras use IT-CCD sensors.

2.8.5 Progressive Scan Interline Transfer CCD (PSIT-CCD).

PSIT-CCD sensors have the same architecture as IT-CCD sensors except that these sensors are able to simultaneously accumulate an image, and output the image sequentially. The result is a non-interlaced image with full vertical and horizontal resolution in a single shutter event, removing the limitations of the IT-CCD.

The PSIT-CCD also frees the user from traditional television resolutions. In comparison to general TV format sensors, the square pixels of PSIT-CCD sensors permit consistent dimensional analysis of an object despite its orientation to the sensor. Additionally, PSIT-CCD sensors output the individual colour of a pixel, thus eliminating the need for additional signal processing to achieve this as opposed to IT-CCD.

2.8.6 FT-CCD versus PSIT-CCD.

Of the five type of image sensors discussed, the FT-CCD and PSIT-CCD sensors are the two most suitable for MV of Froth Flotation. Both sensors have electronic shuttering capabilities and shielded image areas, enabling the sensors to successfully image dynamic scenes. The sensors can achieve the minimum required dynamic range of 48dB, while FT-CCD sensor generally have dynamic ranges larger than 60dB. Both sensor types are capable of frame rates in the order of 25 frames/s and higher, while FT-CCD sensor can typically achieve frame rates of hundreds of frames/s. Both sensors have square pixels, permitting consistent dimensional analysis of an object despite its orientation to the sensor. Access to the individual colour content of each pixel is possible, with no requirement for additional signal processing electronics.

The decisive factor in selection between the two sensors is their cost. The FT-CCD and PSIT-CCD sensors both satisfy the requirements for imaging flotation processes; yet FT-CCD sensors are priced much higher than similar resolution PSIT-CCD sensors.

2.9 Specific Choice of PSIT-CCD Sensor.

After much deliberation, coupled with previous experience with interline CCD sensors; a PSIT-CCD sensor from SONY semiconductors was selected. SONY provides a limited selection of colour and monochromatic PSIT-CCD sensors with effective pixel resolutions ranging from 659 (H) x 494 (V) to 1300 (H) x 1030 (V), with optical formats ranging from 1/4 inch to 2/3 inch. The most attractive property of SONY CCD sensors is that for every colour CCD sensor there is a monochromatic equivalent, making it possible to develop a MVC able to support both a colour or monochromatic sensor, without the requirement of implementing additional hardware to accommodate this. For the implementation of the MVC, the ICX084AK/AL (colour/monochrome) PSIT-CCD sensors were selected.

2.10 The ICX084AK and ICX084AL Sensors.

The ICX084AK and ICX084AL CCD sensor are 1/3-inch progressive scan interline transfer CCD sensors, with a square pixel (7.4μm x 7.4μm) array structure that supports a VGA format. The main feature of the sensors is that all individual pixels can be read within approximately 1/30 second period (30 Frames/s). Furthermore, electronic shuttering with variable charge-storage enables the sensors to realise full-frame still images, without the use of a mechanical shutter. Both sensors have a high resolution (692 H X 504 V) and sensitivity, making them suitable for imaging Froth Flotation processes.

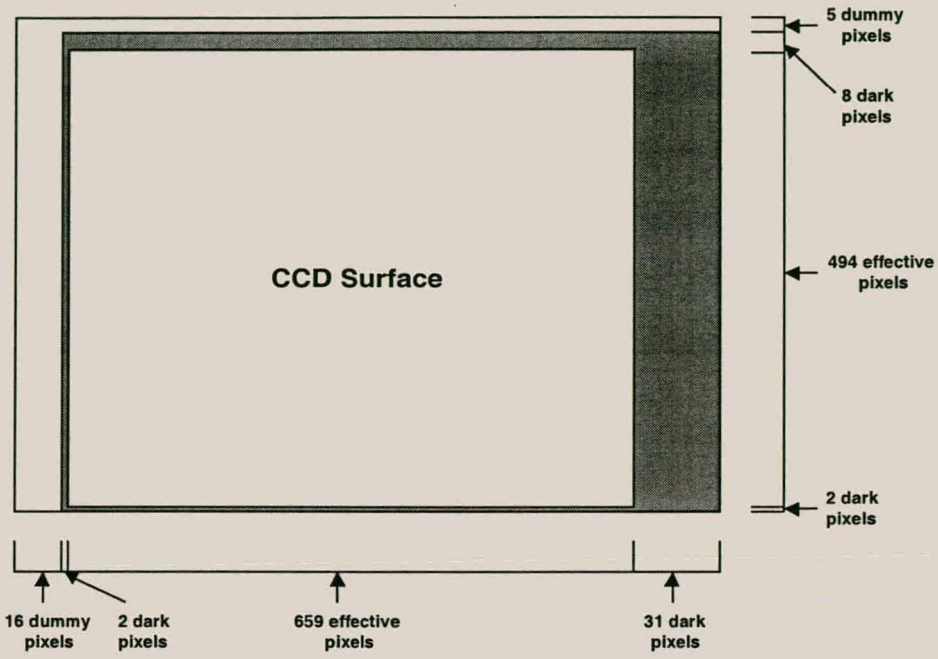

Figure 2.5 Pixel composition of CCD.

Figure 2.5 shows the individual pixel make up of each horizontal and vertical line of the sensors. The function of the dummy pixels is to allow imaging electronics to reach a steady state before valid pixels are read. This will ensure that any unwanted dynamic response from the imaging electronics will not influence valid pixels. Dark pixels are used to calibrate the zero signal reference of the CCD sensor.

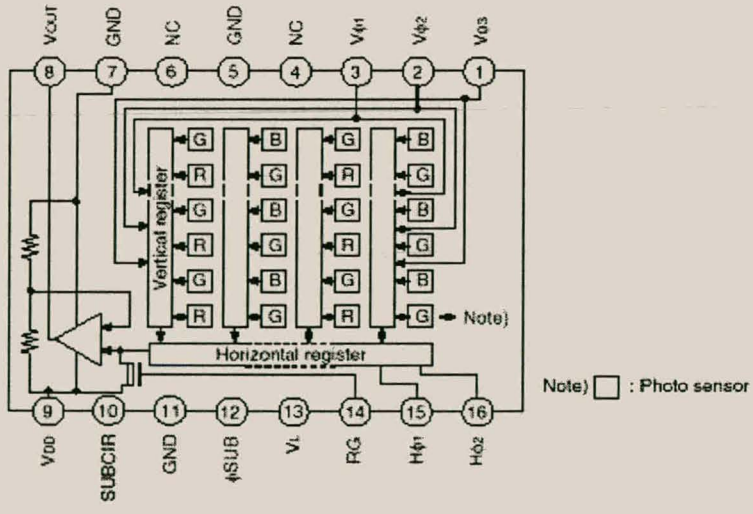

Figure 2.6 Block diagram of ICX084AK CCD.

Figure 2.6 shows a block diagram of the ICX084AK CCD sensor. Light entering the silicon in the image-sensing area of the CCD causes free electrons to be generated and collected in the photocells (pixels). The amount of charge collected in each cell is a linear function of the incoming light intensity and the exposure time. After exposure (controlled by ϕSUB), the charged cells are transferred from the image area to the vertical transfer register, and shifted down the vertical transfer column by the vertical register transfer clocks (Vϕ1, Vϕ2, Vϕ3) to the horizontal register. Each charge is individually clocked out of the horizontal register by the horizontal register transfer clocks (Hϕ1, Hϕ2) to the charge amplifier, which in turn converts each charge to a voltage. After the charge amplifier has converted a charge packet, the amplifier is reset to a rest potential by the reset gate clock (RG). The ICX084AL CCD sensor has the same structure and functioning as the ICX084AK, except that the sensor pixels are sensitive to monochromatic light.

2.11 Characteristics of the ICX084AK/AL CCD sensors.

The previous parts of this chapter were concerned with an evaluation of the most important parameters of a CCD sensor and their relation to imaging a flotation process. These parameters are now constricted, and the ICX084AK/AL CCD sensors will be evaluated accordingly.

2.11.1 Sensitivity and Spectral Response.

It was pointed out in paragraph 2.5 that additional lighting will be used when monitoring Froth Flotation processes with a MVC. This leads to the assumption that since the lighting environment will be constant under the user's control, the sensitivity of a CCD sensor can be fully used. This renders the sensitivity an unimportant parameter in selecting a sensor for imaging flotation. In designing a MVC, the given sensitivity of the CCD sensor is useful to predict the sensors response only if the lighting conditions match those in which the sensitivity of the sensor was measured. Since this will not be the case

for imaging flotation, no specific attempt was made to quantify the sensitivity for the two CCD sensors.

Figure 2.7 shows the relative spectral sensitivity response of each sensor (with test lens) as quoted from data sheets (SONY, 1997a:1090, 1106). Each response curve is normalised, with respect to the individual sensor. The ICX084AK's response indicates that the RGB response of the sensor is well matched. This is an important characteristic, since if a large mismatch between the RGB sensitivities exists; additional hardware signal processing would be required to balance the RGB response of the sensor in order to fully utilize the dynamic range of the sensors. From the two figures it can be seen that the visual spectrum (400nm – 700nm) is suitably covered by the sensors.

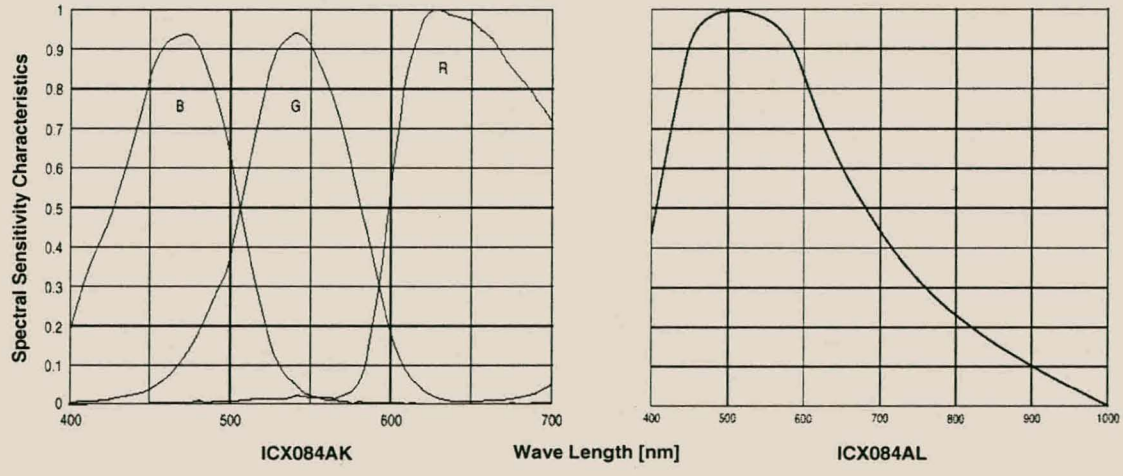

Figure 2.7 Spectral Sensitivity Characteristics of CCD Sensors.

When both sensors are used to image the same pattern under the same illumination conditions (SONY, 1997a:1085, 1102), the data sheets of the sensors show that the minimum sensitivity of the ICX084AL sensor is 50% larger than the sensitivity of the ICX084AK. This sensitivity comparison illustrates that it will be possible to increase the sensitivity (at the cost of losing RGB colour) of the MVC by simply interchanging the two sensors.

2.11.2 Image Sample Rate.

The ICX084AK/AL sensors have a typical frame rate of 30 frames/s (or a data rate of 11.6Mb/s), when using a horizontal drive frequency of 12.2727MHz. To keep the MVC versatile in imaging a variety of Froth Flotation processes while trying to keep the processing requirements of the support electronics (e.g. electronic signal processing and frame grabber) to a minimum; a frame rate of 20 frames/s was selected as the maximum operating speed of the camera. This translates to a maximum horizontal drive frequency of 8.19MHz, or a data rate of approximately 7.7Mb/s.

2.11.3 Resolution and Colour Coding.

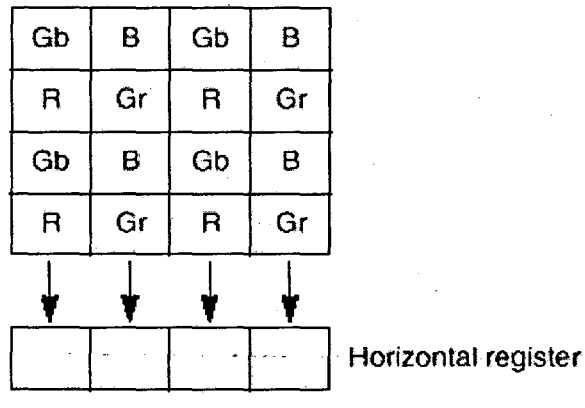

Figure 2.8 Colour coding diagram of the ICX084AK.

Both the ICX084AK/AL sensors have a resolution of 692 (H) x 504 (V) pixels. The main difference between the two sensors is that the ICX084AK sensor uses primary colour filters arranged in a layout as shown in Figure 2.8 to represent the RGB colour of an image pixel. This Filter arrangement for the ICX084AK halves the vertical and horizontal resolution of the sensor to 346 (H) x 252 (V) colour pixels; since the sensor uses four pixels (Gb, Gr, R, B) to represent a colour pixel. The ICX084AL uses a monochrome

filter. Thus, a higher resolution from the MVC can be obtained (at the cost of losing colour) by simply interchanging the two sensors.

Even though the resolution of the ICX084AK sensor is one quarter of the resolution the ICX084AL sensor; Chapter 4 will show that a resolution of 346 (H) x 252 (V) is sufficient to image a typical Froth Flotation process.

2.11.4 Dynamic Range.

It was pointed out in paragraph 2.6, that a CCD sensor with a dynamic range of 48 dB to maximum of 60 dB (8-bit digitisation) should deliver a high image quality, while keeping sensor costs low and making the interface of the camera to a signal processing platform simple and versatile. The maximum expected dynamic range for the ICX084AK/AL sensor will be approximately 56dB (with electronic signal processing: correlated double sampling), which exceeds the minimum required value for 8-bit digitisation. The derivation of the dynamic range for the ICX084AK/AL CCD sensors can be found in Appendix A.

2.12 Conclusion.

After consideration of the imaging capabilities of the various types of CCD sensors available for the custom development of a MVC today, it was found that not all types are suitable to image Froth Flotation processes. Once the forms of image processing and their information requirements were determined, the specific abilities of the CCD sensor could be addressed. This led to the realisation that the sensor type is the primary specification to consider when selecting a sensor to develop a MVC for Froth Flotation, since all the other parameters (spectral response, resolution, sensitivity etc.) are easily satisfied by most commercial and scientific CCD sensors.

Since the dynamic range of most commercial and scientific CCD sensors is typically high (> 40dB) it was decided to limit the bus width of the MVC to 8-bits. This ensures a high degree of compatibility of the MVC with current bus standards and keeps the amount of data from the MVC to a minimum, while still delivering an acceptable image quality.

A compromise between the relative performance and cost of specific type of sensors indicated that PSIT-CCD sensors are suitable sensors; able to achieve high quality images at a fraction of the cost of similar CCD sensors (e.g. FT-CCD). For the implementation of the MVC, the ICX084AK/AL (colour / monochrome) PSIT-CCD sensors from SONY semiconductors were selected. These sensors can be interchanged on the MVC to reduce/heighten the specifications and results achieved accordingly. These sensors satisfy all the requirements need to image Froth Flotation, and will make the development of a MVC that can support both colour or monochromatic images without the need of any additional hardware possible.

CHAPTER 3

IMPLEMENTATION OF THE MACHINE VISION CAMERA.

The focus of this chapter is the implementation of the ICX084AK/AL CCD sensors in a MVC for Froth Flotation. An explanation of the design methodology and goals of the MVC is given, followed by a brief description of the basic structure and components of a typical MVC. A more detailed look at the same components for the MVC as implemented in this thesis is then undertaken, whereafter a simple interface used to transfer images from the MVC to a standard PC for the testing and evaluation of the MVC is presented.

3.1 Design Goals and Methodology for the MVC.

Through discussion of flotation processes in Chapter 1, it was illustrated that a flotation process consists of numerous flotation cells, where each cell will be monitored by a MVC. It cannot be expected that a plant operator or another person continuously control or maintain each camera. Consequently, one of the most important abilities of a MVC is to function as autonomously as possible. With autonomy in mind, the basic design goal of the camera is to capture high quality images of a flotation cell with the minimum amount of interaction required from the user. Excepting the initial setup of the cameras and the occasional cleaning of the camera's lenses, no other physical interaction is foreseen.

In some MVC designs, it may be required to discretely self-implement some of the primary components (signal processing electronics, timing generator etc.) of the camera. This is generally the case when the CCD sensor has no support and signal processing electronics, the required components are not available, or the component costs are too high.

SONY supplies an extensive range of signal processing and support electronics at a lower cost when compared to the self-implementation of these components. Using these support

chips also decreases the development and design time of the camera. Therefore, the design methodology for the MVC is to use the support electronics provided by SONY.

3.2 Basic Structure and Components of a MVC.

Figure 3.1 shows a typical block diagram of a MVC. The six primary components of the camera are the:

- Timing Generator
- CCD clock drive electronics
- CCD sensor
- Signal processing electronics
- Analog to digital converter
- Camera interface electronics

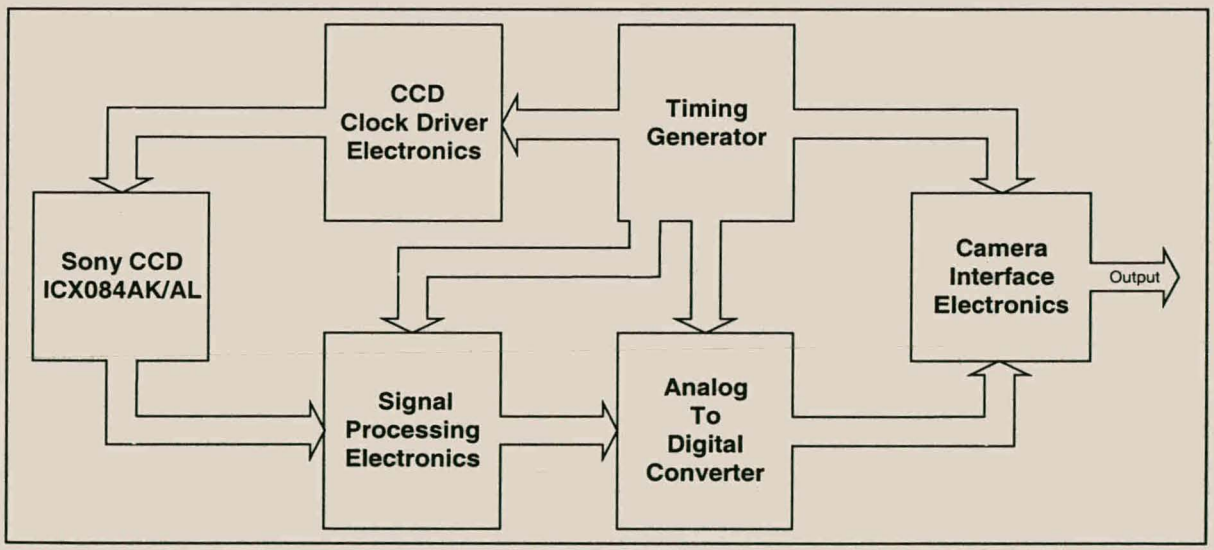

Figure 3.1 General block diagram of the MVC.

The basic principle of the camera operation is as follows: the Timing Generator synchronises and controls the CCD sensor and the other camera subsystems. The CCD sensor continuously outputs analog images, which in turn are amplified and filtered (removal of noise) by the signal processing electronics. The analog-processed image is then converted to a digital format by an Analog-to-Digital (A/D) converter and is output by the interface electronics. Figure 3.2 shows a more detailed block diagram of the MVC as implemented in this thesis. Appendix B contains the detailed schematics for the MVC.

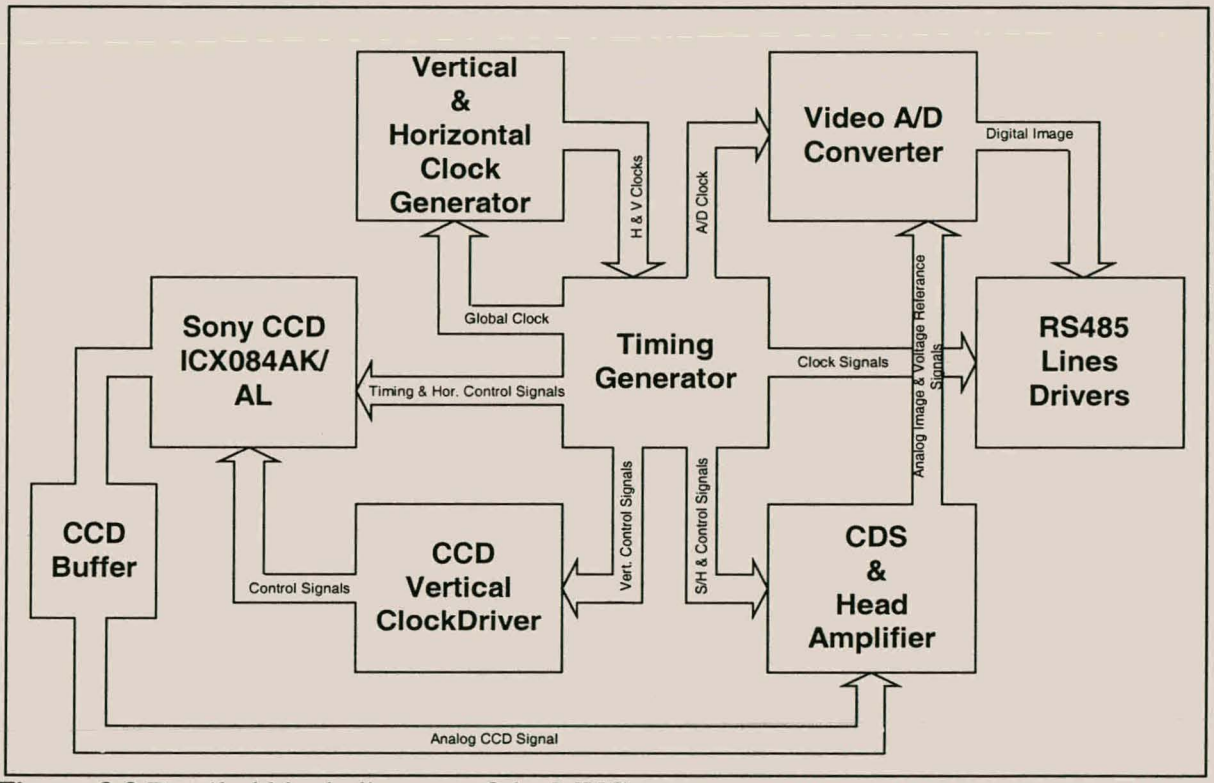

Figure 3.2 Detailed block diagram of the MVC.

3.3 Timing Generator.

The CXD2434TQ Timing Generator of SONY was used, since it is the recommended timing generator for both the ICX084AK/AL CCD sensors and is compatible with all of SONY's signal-processing circuits (SONY, 1997a:1675).

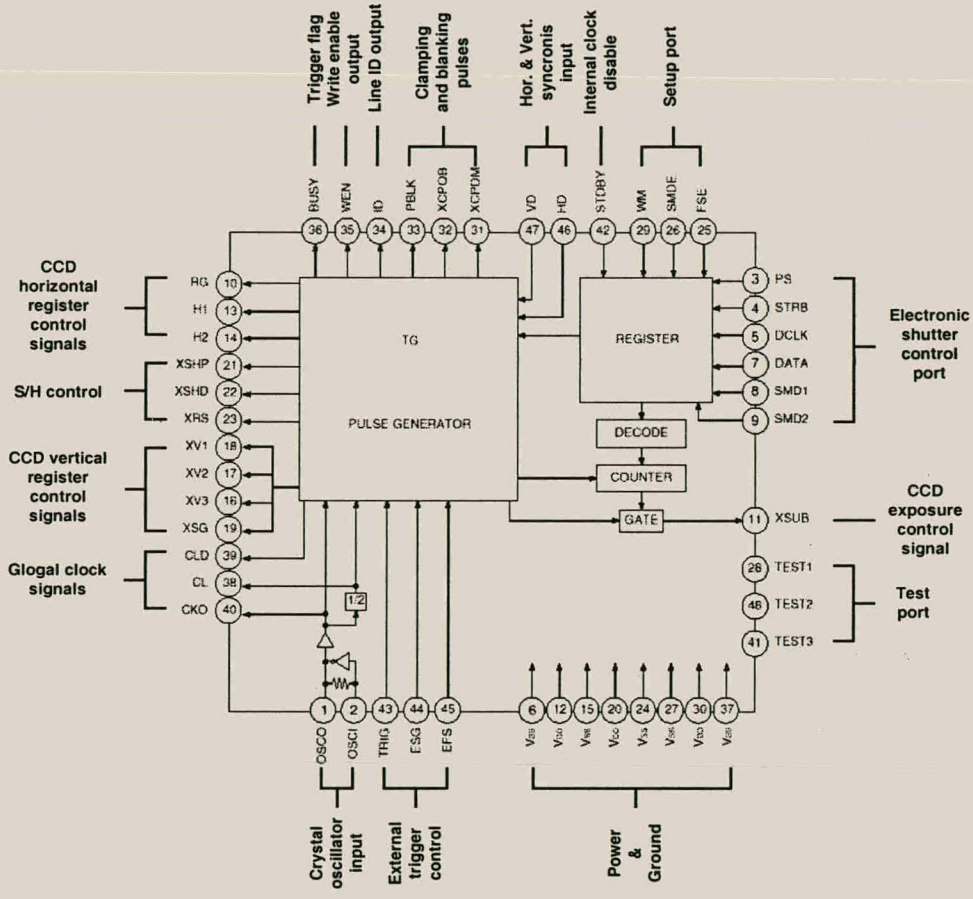

Figure 3.3 Block diagram of CXD2434TQ Timing Generator.

Figure 3.3 shows a basic block diagram of the Timing Generator. The primary functions of the Timing Generator are to

- supply horizontal- and vertical register control clocks; reset gate clock and substrate clock; and control the exposure time and clocking of pixels from the CCD sensor.
- supply sample and hold clocks, and clamping and blanking pulses used by the signal processing circuits (A/D converter, Head amplifier and Correlated double sampler) to synchronise with the pixel output of the CCD sensor.
- supply data valid signals, which can be used by the image-processing platform to detect and capture image data from the MVC.

3.3.1. Exposure Control.

Typically, the Timing Generator is clocked using a crystal with a base frequency of 24.5454MHz. This causes the Timing Generator to clock the CCD sensor to a maximum image rate of 30 frames/s. To achieve the required image rate of 20 frames/s as discussed in Chapter 2; the Timing Generator must be clocked with a 16.38MHz crystal.

The Timing Generator provides two methods in order to control the time at which an image from the CCD sensor is output. The first method is through the External trigger mode of the Timing Generator. This allows the user the randomly decide when an image must be clocked out of the sensor. The second method is to let the Timing Generator continuously self-clock images out of the sensor.

The second method is preferred over the first, since it will provide a sufficient exposure (integration) period of 50ms for the CCD sensor and requires no decision from the user regarding when to capture images from the MVC. In principle, the same functionality could be implemented with the External trigger function, but this would require additional hardware in order to make it autonomous of the user (Chapter 4.3.2). To enable the second method, the electronic shutter function (exposure control) of the Timing

Generator is simply disabled (SONY, 1997a:1685). This is done by setting PS, SMD1 and SMD2 to High.

3.3.2 Detection of Valid Image Data.

The Timing Generator supplies two control lines; Write enable (WEN) and Line Identification (ID), to indicate when the CCD sensor is outputting image data. Depending on the initial setup of the Timing Generator; either the WEN line alone or both the WEN and ID line are required to determine when valid pixel data is being output. Figure 3.4 shows the switching of the two control lines, with the electronic shutter of the Timing Generator disabled.

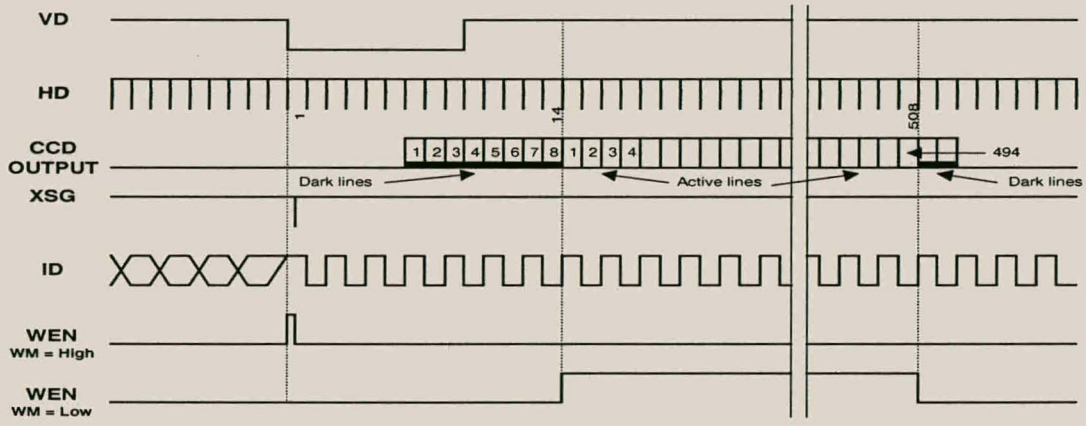

Figure 3.4 Functional timing diagram of WEN and ID control lines.

The configuration of WEN is controlled by setting the WEN mode line (WM) to High or Low. If WM is High, the WEN line is synchronised with the CCD sensor charge readout pulse (XSG) and both the WEN and ID lines will be required to capture valid image data. By setting WM to low, the WEN signal functions as a write valid signal and is only High when active image lines are being output by the CCD sensor. For the implementation of the MVC only the WEN signal is used, since this method is the least hardware intensive for capturing image data. Additional information concerning the Timing Generator can be found in the relevant data sheets (SONY, 1997a:1675-1701) and the MVC schematics in Appendix B.

3.4 Vertical and Horizontal Clock Generator.

The Timing Generator requires a Horizontal- (HD) and Vertical (VD) synchronous input signal. These signals are used by the Timing Generator to synchronise the drive and control pulses for the CCD sensor.

Figure 3.5 shows the functional timing of HD relative to the secondary clock (CL) of the Timing Generator, while Figure 3.6 shows the functional timing of VD relative to HD. During every cycle of the VD, an image is output by the CCD sensor, while every HD cycle coincides with the output of a horizontal line of an image.

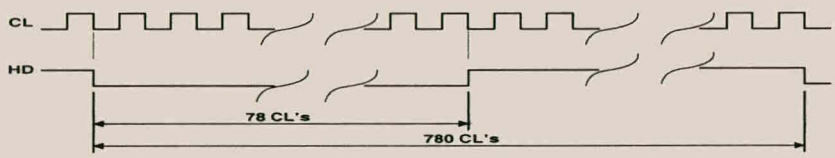

Figure 3.5 Functional timing of HD relative to CL.

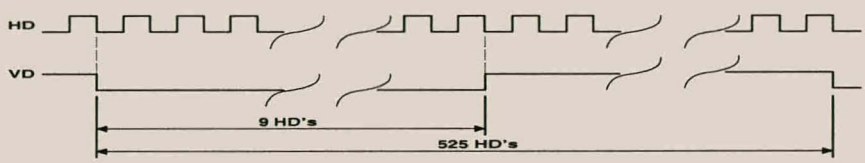

Figure 3.6 Functional timing of VD relative to HD.

To generate the HD and VD signals, an EPM7064SLC44-10 CPLD of ALTERA was used. The HD en VD pulse are derived from the CL clock, which runs at half the frequency of the base clock (CKO) of the Timing Generator. The data sheets of Timing Generator specify that VD and HD must have a minimum setup time of 20ns before CL goes Low, and must have a minimum hold time of 5ns (SONY, 1997a:1681).

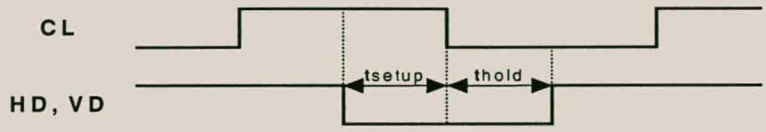

Figure 3.7 Phase conditions of HD and VD relative to CL.

Figure 3.7 shows a timing diagram of the required phase conditions of HD and VD relative to CL. By keeping the design of the CPLD positive edge triggered with respect to CL (running at 16.38/2 MHz) the phase conditions of HD and VD could easily be achieved with typical setup- and hold times of 56ns and 66ns. The VHDL source code of the CPLD design can be found in Appendix C. ALTERA (1996) provides additional information about the CPLD.

3.5 Vertical Clock Driver.

The Timing Generator generates the vertical transfer register clock signals, horizontal transfer register clocks signals, reset gate clock signal and the substrate clock signal (CCD exposure signal) for the ICX084AK/AL CCD sensor. Unfortunately, only the horizontal transfer register clock signals can directly interface with the CCD sensor, while the reset gate clock signal can be capacitively interfaced with the CCD sensor. The vertical transfer register- and substrate clock signals have to be converted to a set of signals with the correct voltage bias levels and polarities to interface with the CCD sensor.

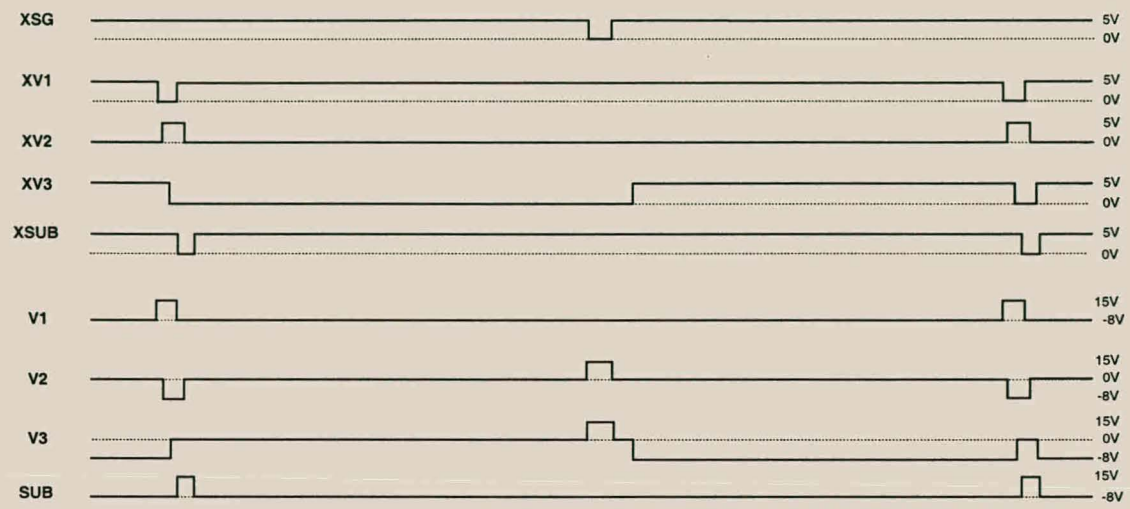

Figure 3.8 Functional read-out timing diagram of the CCD sensor.

Figure 3.8 shows a functional readout timing diagram of the vertical transfer register (XV1, XV2, XV3)- and substrate clock (XSUB) signals of the Timing Generator and the corresponding set of signals required by the CCD sensor (V1, V2, V3, SUB) (SONY, 1997a:1090). To convert these signals from the Timing Generator to the signals indicated in Figure 3.8, the CXD1267AN vertical clock driver from SONY is used. The clock driver is also required, since the Timing Generator is incapable of driving the high capacitive loads of the vertical transfer register clock inputs of ICX084AK/AL CCD sensors. Table 3.1 shows the truth table for the clock driver.

Table 3.1 Truth table for the CXD1267AN vertical clock driver.

Input				Output		
$V_{IN}1,3$	$SG_{IN}1,2$	$V_{IN}2,4$	SHT_{IN}	$V_{CCD}1,3$	$V_{CCD}2,3$	$V_{CCD}SHT$
L	L	X	X	V_H	X	X
H	L	X	X	Z	X	X
L	H	X	X	V_M	X	X
H	H	X	X	V_L	X	X
X	X	L	X	X	V_M	X
X	X	H	X	X	V_L	X
X	X	X	L	X	X	V_H
X	X	X	H	X	X	V_L

X: Don't care; Z: High impedance; V_H: 15V; V_M: 0V; V_L: -8V

Table 3.1 and Figure 3.8 illustrate that by connecting XSUB to SHT_{IN}, the required VSUB signal is created. Similarly, by connecting XSG to $SG_{IN}1$ and $SG_{IN}2$, XV1 to $V_{IN}2$, XV2 to $V_{IN}1$, and XV3 to $V_{IN}3$ the required V1, V2 and V3 signal are created.

After the vertical transfer register- and substrate clock signals of the Timing Generator have been adjusted by the clock driver, a maximum propagation delay of 200ns can be expected for these signals (SONY, 1997a:1743). This delay will have no effect on the performance of the CCD sensors, since there is sufficient time (maximum of approximately 1.47μs[1]) between horizontal register transfers to compensate for the propagation delays of V1, V2, V3 and SUB (SONY, 1997a:1092, 1108). Further information about the clock driver can be found in the relevant data sheets (SONY 1997a:1740-1747).

3.6 Correlated Double Sampler (CDS) and Head Amplifier.

It was noted in Chapter 2 that in order to ensure the maximum Dynamic Range of 56dB for the ICX084AK/AL CCD sensors is used, Correlated Double Sampling is required. Additional signal processing is also required in order to match the output voltage range of the CCD sensor to the input voltage range of the A/D converter. The 8-bit resolution of the A/D converter is thus fully utilised.

[1] When driving the Timing Generator with a 16.38MHz crystal.

3.6.1 Correlated Double Sampling (CDS).

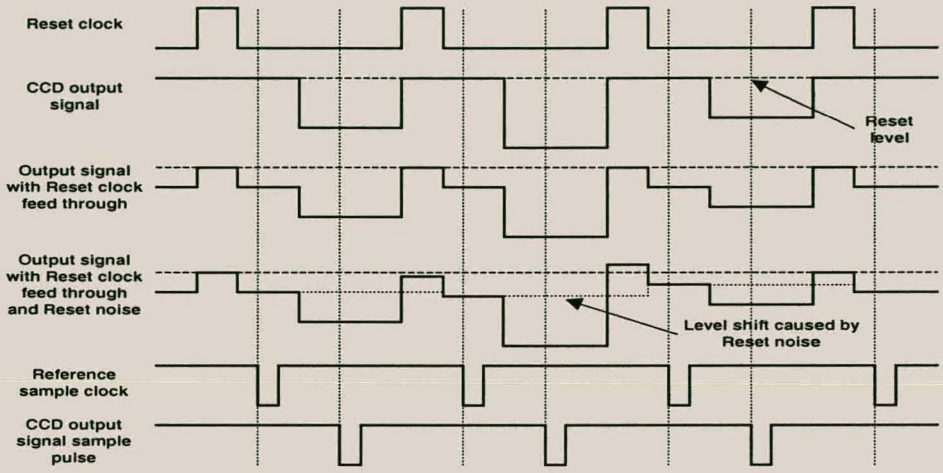

Figure 3.9 Functional timing diagram for CDS.

Figure 3.9 shows the basic principle of CDS. During every period of the output signal from the CCD sensor, the reset- and charge levels of the signal are consecutively sampled. Any noise fluctuations common to both samples will thus be eliminated, making CDS an efficient method to eliminate fixed (e.g. Reset clock feed through) and low frequency noise (e.g. Reset noise, Dark signal) from the output signal of the CCD sensor. More information about sensor noise can be found in Appendix A.

3.6.2 On-Chip Analog Signal Processing.

SONY supplies various multi-purpose head amplifiers for digital CCD cameras, of which the CXA1690Q is the most useful. This device provides a high gain signal amplifier with automatic gain control for CCD sensor signals; a chromatic and line signal amplifier; blanking; CDS; a sample and hold function, and reference voltages for A/D converters.

3.6.2.1 Functional Description of the CXA1690Q.

Figure 3.10 shows a basic block diagram of the CXD1690Q. By setting CAM/VIDEO and PB/REC, the device can be placed in camera system or video system mode. The video mode supports TV format signals, while the camera mode is used to directly interface with CCD output signals.

For the implementation of the CCD camera for Froth Flotation, the camera system mode is used. To Correlate Double Sample and to calibrate the black reference of the CCD output signal, the CXD1690Q uses the sample and hold, and clamping pulses of the Timing Generator. When interfaced with the A/D converter the Automatic gain control (AGC) CCD signal amplifier is used to ensure that the maximum Dynamic Range of the CCD sensor is exploited. The AGC amplifier is able to adjust both the gain (through AGCMAX and AGCCONT) and offset (OFFSET) of the resultant CCD signal to the A/D converter.

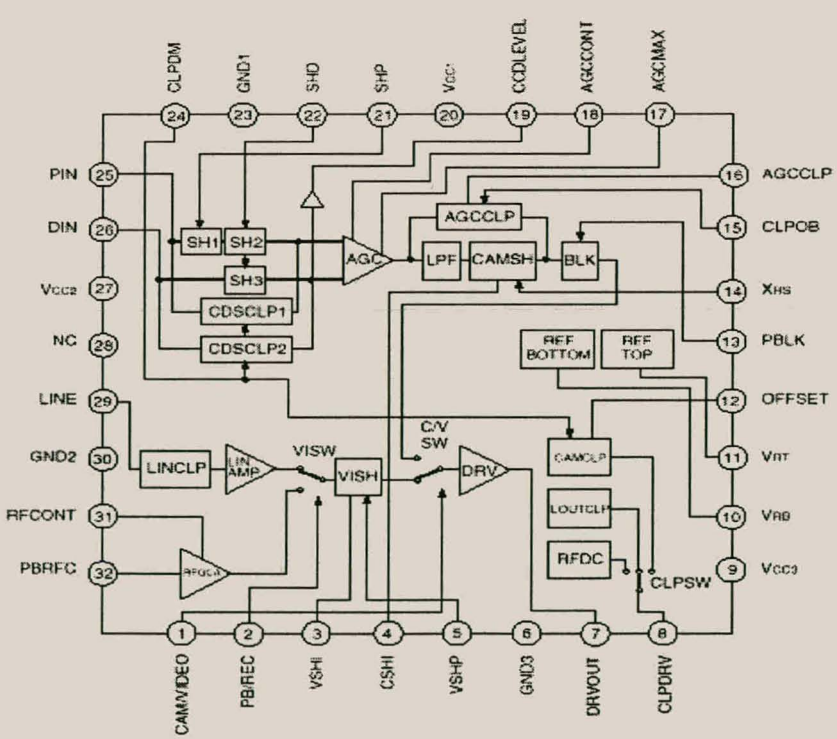

Figure 3.10 Basic block diagram of the CXD1690Q.

3.6.2.2 Correlated Double Sampling with the CXD1690Q.

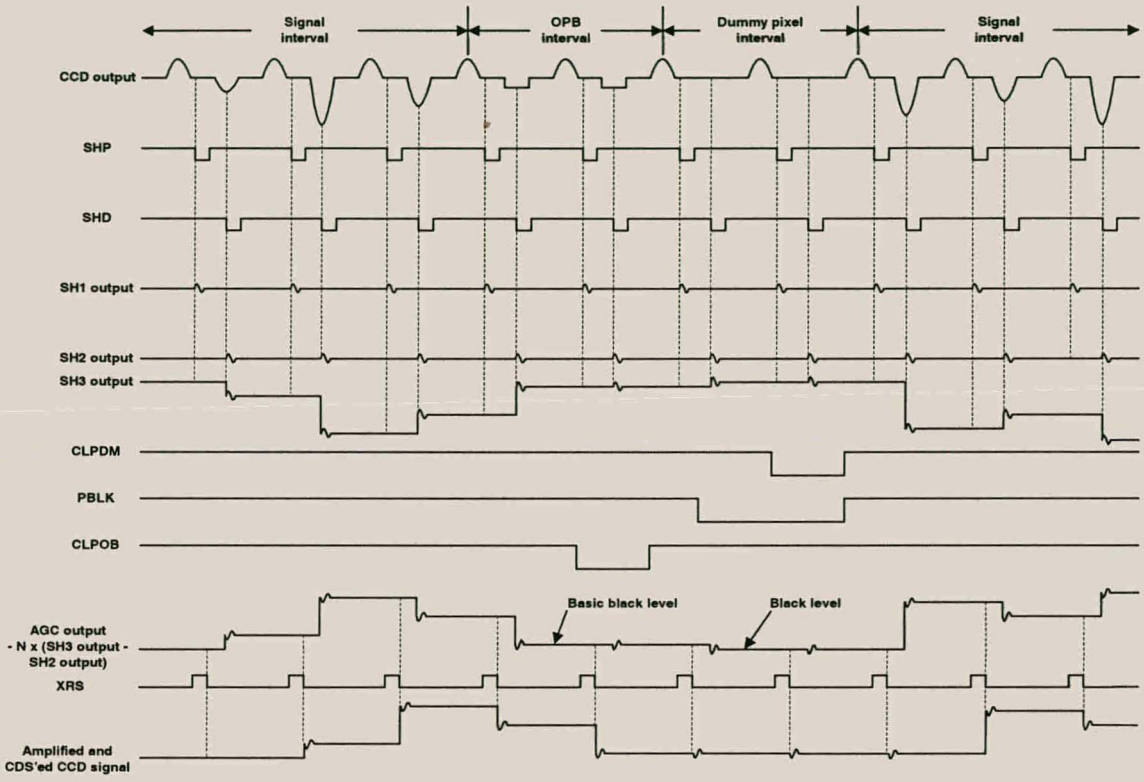

Figure 3.11 Reduced functional timing diagram of the CXD1690Q.

Figure 3.11 shows a reduced timing diagram of the CXD1690Q. The analog output of the CCD sensor enters PIN and DIN; each through a 1µF capacitor where it is Correlated Double Sampled. The reset potential, followed by the analog value of the pixels is sampled by the SHP and SHD signals (supplied by the Timing Generator) to create the SH1, SH2 and SH3 signals.

The DC level of the SH2 and SH3 signals are initially equalised during the dummy pixel interval of the CCD sensor (CLPDM), by the CDS clamps (CDSCLP1 & 2). The signals are then sent to the AGC differential amplifier (Figure 3.10). This will ensure that any input offset to the AGC amplifier is eliminated.

The AGC then amplifies the difference between the SH2 and SH3 signal by a preset gain; and the resultant signal passes through a Low-pas filter (LPF) to minimise high frequency noise. The signal to noise ratio (S/N) of the amplified signal is thus improved.

The Camera sample and hold circuit (CAMSH) samples the output of the LPF to synchronise the output signal of the LPF with the falling edge of the XRS signal from the Timing Generator. This will ensure that the output signal from the LPF is optimally synchronised with the A/D conversion clock (CLD) of the Timing Generator. A more detailed timing diagram of the CXD1690Q can be found in the relevant data sheets (SONY, 1997a:1890).

3.6.2.3 Black Reference.

The black reference, also known as the black level, is an integral part of a CCD output signal since it is used to calibrate the support electronics of the CCD sensor (CDS and A/D converter). This allows for successful distinction between black- and valid pixel data from the CCD sensor.

To set the black reference, the Timing generator provides two signals. The first is an optical black clamping period (CLPOB) signal to indicate when dark pixels are being output by the CCD sensor. A second blanking signal (PBLK) indicates when to set the black level. During the dark pixel period, the CXD1690Q uses CLPOB to detect the black level by clamping the output of the amplifier during the black pixel period of the CCD sensor. Final adjustment of the black reference is made by adjusting the black level of the AGC amplifier during the dummy pixel interval of the CCD sensor, (when PBLK is low), to ensure that a change in the black level does not influence valid pixel data.

To allow camera signal levels to stabilise before valid pixels are sampled by the A/D converter, the CXD1690Q provides an output clamp signal (CLPDRV) which clamps the output of the CXD1690Q (DRVOUT) during the dummy pixel interval of the CCD sensor. This is done while the black level is being set (PBLK is low). The output clamp

signal also contains an offset voltage control input (OFFSET), which allows for the adjustment of the DC level of the signal output from the CXD1690Q. By using the OFFSET input, the DC level of DRVOUT can be set to the required 2V level for the A/D converter.

3.6.2.4 Automatic Gain Control (AGC) Amplifier.

The linear AC output voltage of the ICX084AK/AL CCD sensors varies between 0mV to –500mV, while the input voltage range of the A/D converter of the MVC is between 2V to 4V. The DC bias level of the output signal of the CCD sensor must therefore be shifted to 2V, and amplified by a minimum factor of -4 in order to make full use of the 8-bit resolution of the A/D converter.

The AGC amplifier provides a maximum gain ranging from 19 to 43dB, allowing the MVC to function under various illumination conditions. The gain is controlled by varying the bias voltages of the AGCMAX and AGCCONT input pins of the CXD1690Q (Figure 3.10). For the implementation of the MVC for Froth Flotation, a decision was made to allow the gain to be adjusted manually to a preset value. Automatic gain control was considered to be unnecessary; since once the MVC has been set up and adjusted for maximum Dynamic Range, no additional adjustment is necessary.

3.6.2.5 Voltage Reference Signals for the A/D Converter.

The CXD1690Q provide a 2V and 4V, lower (V_{RB}) and upper (V_{RT}) voltage reference for A/D converters. The CXD1690Q is able to drive a minimum V_{RB}-V_{RT} load of 160Ω. The A/D converter used for the MVC has a minimum rating for its V_{RB}-V_{RT} load of 230Ω. Thus, the reference voltages from the CXD1690Q can directly interface with the A/D converter.

3.7 A/D Converter.

In all digital cameras, a high-speed A/D converter is required to convert the analog CCD signal from the CXD1690Q into a digital format for storage. For the development of the MVC, the CXD1175AM CMOS A/D converter from SONY was selected. This device has a TTL compatible output; low power consumption; 8-bit ± ½-bit resolution; and a maximum conversion rate of 20MHz (SONY, 1997b).

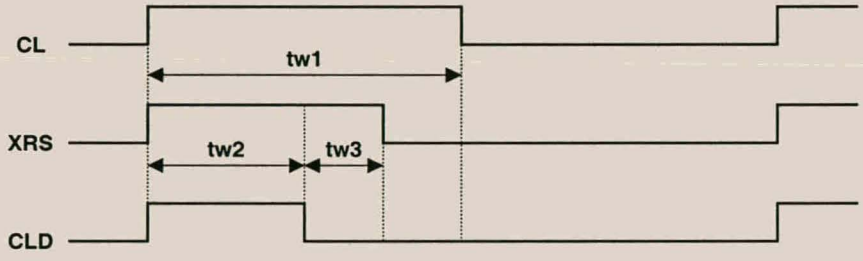

Figure 3.12 Relative timing of CL, XRS and CLD.

Discussion of the Timing Generator (paragraph 3.3) showed that this device provides sample and hold-, and A/D conversion signals to support additional signal processing devices. Figure 3.12 shows the relative timing of the secondary clock (CL); synchronization clock (XRS); and the A/D conversion clock (CLD) of the Timing Generator.

The timing of XRS and CLD is arranged in such a manner that the output of the AGC amplifier of the CXD1690Q is sampled a minimum of tw3 (ns) before it changes. This will ensure that the output of the CXD1690Q is sampled at its most stable state, minimising the effects of clock feed through and other dynamic responses in the output signal from the CXD1690Q. The time (tw3) that the XRS pulse is longer than the A/D conversion pulse may vary from 0ns to 15ns (typically 8ns) for 30 frames/s image rate. Since a 20 frames/s rate will be used tw3 will vary from a minimum of 0ns to 22ns (typically 12ns). In practice, tw3 will always be larger than 0ns, since after XRS falls the output signal from the CXD1690Q does not change immediately.

The A/D converter requires that the positive pulse width of CLD (tw2) must be a minimum of 25ns. The time tw2 of CLD can vary from 17ns to 27ns (typically 22ns) for a 30 frames/s image rate. For a 20 frames/s rate, this time will vary from 25.5ns to 40ns (typically 33ns), satisfying the minimum requirement of 25ns.

Figure 3.13 shows the timing for the A/D converter. The A/D converter uses a 2-step conversion method, which causes a delay of 2.5 A/D clock periods for 50% duty cycle A/D conversion clock. Data is sampled by the falling edge of CLD, and digital data is output on the rising edge of the third CLD clock with an additional output delay of Td. To capture the digital output of the A/D converter it was decided not to use CLD but CL, since CL has a 50% duty cycle and the output of the A/D converter is valid during the falling edge of CL.

For CL to be a suitable data clock to capture data from the A/D converter, tw1 ≥ Td. For a maximum frequency of 8.19MHz for CL, the minimum width of tw1 is 57ns, which is larger than the maximum value of 30ns for Td.

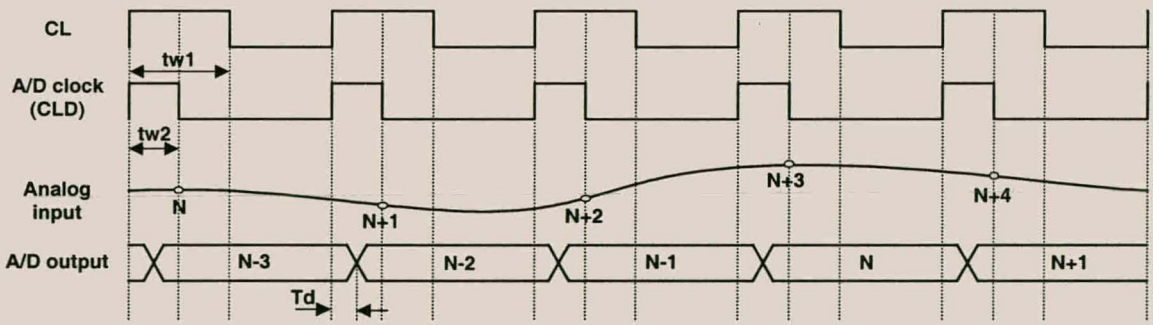

Figure 3.13 Timing diagram for the CXD1175A.

3.8 CCD Output Buffer.

To buffer the signal output of the ICX084AK/AL CCD sensors, a 2N5484 high bandwidth N-Channel JFET in a source-follower configuration is used. The 2N5484 has a high unity gain frequency of 400MHz and low maximum input capacitance of 5pF, making it suitable to buffer the signal from the CCD sensor. Siliconix (1982) provides detailed information about the 2N5484 JFET.

3.9 RS-485 Drivers.

Installation of a machine vision system is often very difficult, since physical limitations regarding the positioning of the MVC relative to the image processing platform are to be expected. To partially relieve this problem, RS-485 drivers were used as the interface standard for the MVC. The RS-485 is a well-known high-speed differential data communication standard, suitable for use in electronically noisy environments, such as Froth Flotation plants. For a maximum frame rate of 20 frames/s (approximately 7.7Mbytes/s) a cable of no longer than 13m may be used between the MVC and processing platform (Nelson, 1995).

For the implementation of the MVC, three DS96F172 quad differential drivers were used to drive the 8-bit data from the A/D converter, the data valid signal (WEN) and secondary clock (CL) of the Timing Generator to the image processing platform. More information on the DS96F172 quad differential driver can be found in the data sheets of this device (National, 1996).

3.10 Power Supply and Power Consumption.

Table 3.2 Summary of voltage requirements of the components of the MVC.

Devises	+15V	+5V	-8V	Max. ΔV
CXD2434TQ (Timing Generator)		•		±250mV
CXD1175AM (A/D Converter)		•		±250mV
EPM7064S (CLPD)		•		±250mV
CXA1690Q (CDS and Head Amplifier)		•		±500mV
DS96F172 (RS-485 Drivers)		•		±250mV
CXD1267AN (Vertical Clock Driver)	•		•	±450mV (+15V) ±1V (-8V)
ICX084AK/AL (Colour and Monochrome CCD sensor)	•		•	±450mV (+15V)

The components of the MVC require three different supply voltages and are summarised in Table 3.2. Of the above listed devices; the Timing Generator, CPLD, RS-485 drivers and A/D converter place the highest constraint on their +5V supply voltage, which must not vary by more than 250mV. To implement the required voltage sources, the LM78XX/79XX positive and negative linear voltage regulator series are used. These regulators have a line and load regulation of lower than 150mV and are capable of driving a current in excess of 1A, which exceeds the requirements of the MVC. Another important factor for the choice of linear regulators is that these devices typically cause a lower level of noise than switch mode regulators.

A LM7815 regulator is used to create the +15V supply. The resultant +15V supply is fed to a LM7808 and LM7805 regulator to create the +5V supply. A LM7908 is used for the –8V supply. The use of a LM7808 and LM7805 regulator is to minimise the power dissipation of the regulators over the case when only a LM7805 is used to implement the +5V supply.

Table 3.3 Maximum power consumption of the MVC.

Supply to MVC	Current	Power
+18V	330mA	6.0W
-11V	16mA	180mW

Table 3.3 shows the maximum power consumption of the MVC when tested at a 20 frames/s image rate. From this table, it can be seen that the maximum total power consumption of the MVC is 6.2W at an image rate of 20 frames/s. The majority of the power need by the MVC is dissipated as heat in the linear power supply of the MVC.

Table 3.4 Maximum power dissipation for the power supply circuit of the MVC.

Supply	Power dissipation
+15V	1W
+8V and +5V combination	3.3W
-8V	50mW
Total	4.35W

Table 3.4 summarises the maximum power dissipation for the linear regulators. From this table, it can be shown that approximately 70% of the power required by the MVC is dissipated as heat. As a result, the power efficiency of the MVC will be approximately 30%. If lower power consumption is required from the MVC, the linear regulators can be replaced with switch mode regulators. The cost of this is the possible introduction of additional noise to the imaging electronics.

3.11 A Simple Interface for the MVC.

To capture images from the MVC, a simple MVC interface (MVCI) is required. The primary function of the MVCI is to evaluate the performance of the MVC, and to function as a post-processing tool with limited real-time capabilities for the further development of froth feature extraction algorithms.

For the implementation of the MVCI, a standard PC is used as the image-processing platform for the MVC. The MVCI is simply a card that slots into the ISA (Industry Standard Architecture) Bus of the PC. The card is capable of capturing images from the MVC at a rate of 20 frames/s (approximately 7.7Mbytes/s). Captured images are transferred through the ISA Bus to the memory of the PC for image processing.

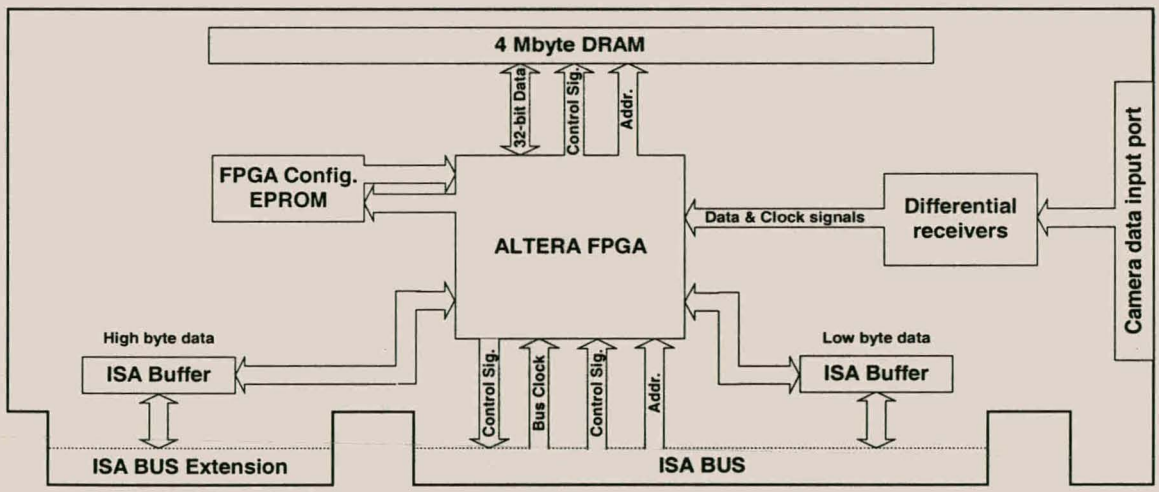

Figure 3.14 Block diagram of the MVC Interface card for a PC.

Figure 3.14 shows the implementation of the Interface card. From paragraph 3.3.1 it is known that the MVC requires no control signals to function and will run autonomously from the MVCI card. Differential image data and clock signals are continuously transmitted to the MVCI by the MVC, and are converted back to TTL levels by the differential receivers. The Field Programmable Gate Array (FPGA) writes the images from the MVC to DRAM for storage, or reads images from the DRAM and downloads

them through the ISA Bus. The Buffers are used to drive data between the FPGA and ISA Bus.

The current implementation of the MVCI card has 4Mbytes of DRAM on board allowing 10 full images to be consecutively captured from the MVC before an image has to be downloaded to the memory of the PC. Images can be downloaded at a typical rate of 6.5 images/s from the MVCI. Appendices D to G contain detailed information about the implementation of the MVCI card.

3.12 Conclusion.

In this chapter the ICX084AK/AL CCD sensors were used to implement a MVC for Froth Flotation. The basic structure of a typical MVC was shown and used as a basis for the development of a MVC for Froth Flotation. The design goal and methology was to make the MVC as autonomous as possible and use available "building blocks" to implement most of the sub-systems of the camera.

The MVC was designed to run autonomously at an image rate of 20 frames/s with manual gain control, to ensure that the maximum Dynamic Range of the MVC can be utilised. Additionally, RS-485 drives where used to ensure that the MVC and its image-processing platform can be separated from one another in order to avoid any practical limitations in the physical positioning of the MV system. No frequent actions from the user are required to operate the MVC, except for the initial set-up and occasional cleaning of the camera lens. The initial set-up itself will only involve the positioning, focusing and configuring of the camera gain.

Through the use of a simple MVCI, images from the MVC can also be captured and evaluated. The MVCI can also help with the development of additional froth feature extraction algorithms.

CHAPTER 4

TESTING AND EVALUATION OF THE MVC.

To evaluate whether the MVC is able to capture suitable images of surface froth from a flotation cell, three tests were implemented. These tests were based on the requirements as discussed in Chapter 2 of a CCD sensor to detect bubbles of a dynamic surface froth.

The first test was chosen in order to evaluate the ability of the MVC to detect objects with dimensions of 1cm. Images of a standard optometrist eye pattern were captured, and evaluated to determine if the MVC will be able to produce suitable images of surface froth with bubbles of 1cm or larger. The second test was designed in order to determine if the smear expected from the MVC is sufficiently low to produce images of mobile surface froth. Images of the same object moving at different speeds are captured at a rate of approximately 20 frames/s and inspected to evaluate the smear of the MVC. The third and final test evaluates the overall maximum SNR (Dynamic Range) of the MVC.

To implement the above-mentioned tests, the MVCI card described in Chapter 3 was used. The frame rate of the MVC was set to 19.51 frames/s, since a 15.98MHz crystal was used from the Timing generator of the MVC. The results of these tests form the bulk of analysis in this chapter.

4.1 Selecting Suitable Optics for the MVC.

The optics proposed to image flotation cells are standard C-mount fixed focal length compound TV lenses, with manual iris and variable focus. These types of lenses are very common, and are standardised with respect to their mounting mechanism and deliver adequate image quality suitable for imaging flotation.

Paragraph 2.7 mentions that the typical areas of flotation cells of which images are taken, are limited to 0.2m x 0.2m to 0.3m x 0.3m for imaging distances ranging between 1m to 2m. These areas require the MVC to resolve a sufficient amount of pixels over a 1cm or larger bubble. The actual pixel resolution requirement (resolving power) for a 1cm bubble is based on the information extracted from the bubble. Since for flotation it is typically only necessary to capture images with sufficient overall detail, and not to extract individual attributes of structures of the surface froth; the resolution for a 1cm bubble must only ensure that the overall image is sufficiently sampled.

Table 4.1 summarises the expected areas to be imaged, and the expected pixel resolution for a 1cm bubble; when imaged from a distance of 1- to 2m using the monochromatic and colour version of the MVC respectively. The results for the table were calculated using the formulae and characteristics for thin-lenses (RCA, 1974:209).

Table 4.1 The expected resolution of a 1cm bubble imaged from 1- to 2m.

Image Destines (m)	Focal Length (f in mm)	F/number (F#)	Imaged Area (H x V in m)	Mono. CCD Resolution (pixels)	Colour CCD Resolution (pixels)
1	21	≤ 9.33	0.23 x 0.17	29.0	14.5
	23	≤ 9.33	0.21 x 0.16	31.8	15.9
	25	≤ 9.33	0.19 x 0.14	34.7	17.3
	27	≤ 9.33	0.18 x 0.13	37.5	18.7
	29	≤ 9.33	0.16 x 0.12	40.4	20.2
1.5	21	≤ 9.33	0.34 x 0.26	19.2	9.6
	23	≤ 9.33	0.31 x 0.23	21.0	10.5
	25	≤ 9.33	0.29 x 0.22	22.9	11.5
	27	≤ 9.33	0.27 x 0.20	24.8	12.4
	29	≤ 9.33	0.25 x 0.19	26.6	13.3
2	21	≤ 9.33	0.46 x 0.34	14.3	7.2
	23	≤ 9.33	0.42 x 0.31	15.7	7.9
	25	≤ 9.33	0.39 x 0.29	17.1	8.6
	27	≤ 9.33	0.35 x 0.27	18.5	9.2
	29	≤ 9.33	0.33 x 0.25	19.9	9.9

Only lenses with focal lengths in the range from 20mm to 30mm were selected, since longer focal length lenses are too bulky and expensive. Shorter focal length lenses will not provide adequate resolving power to image 1cm froth bubbles from a distance of 2m. The maximum value for the F/number listed in Table 4.1 was determined to ensure that the diffraction limit of the lens will allow the expected resolutions of Table 4.1 to be obtained (RCA, 1974:215). The F/number limit also gives an indication of the adjustment range of the lens aperture and is shown by the following equation:

$$F\# = \frac{f}{D} \qquad (4.1)$$

where f is the focal length of the lens (mm) and D is the diameter of the lens aperture (mm).

From Table 4.1 it can be seen that a compromise between the imaged area and resolving power of the MVC must be made; since the higher the resolving power of the MVC, the smaller the expected image area will be. Since the resolving power is the more important parameter, the choice of lens is primarily based on this parameter, while still delivering a reasonable image area. The table also shows the both the monochromatic and colour version of the MVC will provide a high resolution for an 1cm bubble when imaged from a distance of 1m, and degrades linearly for an increase in imaging distance. For the evaluation of the MVC, a F# 1.4 25mm focal length lens was selected. This lens provides a reasonable resolving power and image area for distances ranging from 1- to 2m. The Electro-Optics handbook (RCA, 1974) provides additional detail regarding lenses and their characteristics.

4.2 Evaluation of the Resolving Power of the MVC.

To practically evaluate whether the MVC will be able to image 1cm froth bubbles and obtain the predicted resolution for a 25mm lens; images of a standard eye test pattern were captured. Only the colour MVC is evaluated, since the colour MVC produces

images of a lower resolution than the monochromatic version, and will therefore set the lower boundary of the capabilities of the MVC.

Figures 4.1 to 4.4 show both horizontal and vertical images of the test pattern imaged from a distance of 2m. Figures 4.1 and 4.3 are the raw colour images of the eye pattern when all the RGB colour pixels from the MVC are considered to be monochromatic. This format of the images has the same resolution as to be expected from the monochromatic version of the MVC. Figures 4.2 and 4.4 on the other hand, are the actual RGB colour images from the MVC converted to a monochromatic format. The reason for taking images of the eye pattern from both a horizontal and vertical direction was to show that since the pixels of the MVC are square, the image quality is rotation invariant.

The size of the text between the two bars of the eye pattern is approximately 11mm and 9mm high, with detail smaller than the dimensions of the text. From Figures 4.1 and 4.3 it can be seen that this text is clearly visible with minimal distortion, while even the smaller text on the lower half of the eye pattern can still be clearly identified.

In practice, the true monochromatic MVC will produce higher quality images for the same eye pattern when imaged under the same condition, since the monochromatic CCD sensor has a higher sensitivity and resolution than the colour sensor. From the monochromatic colour images of the eye pattern (Figure 4.2 and Figure 4.4), it can be seen that the same two lines of text are still clearly visible, while the smaller text becomes more difficult to read.

After closer inspection of the two lines of text (using an image processing program e.g. Paint Shop Pro), it was found that the expected and predicted resolution for the 11mm and 9mm text of the eye pattern compared fairly well. Table 4.2 summarises these results. The larger values measured are attributed to the tendency of CCD sensors to spread a high contrast boundary over more than one pixel.

Table 4.2 Summary of the predicted and measured pixel resolution for the 9mm and 11mm text of the eye pattern using a F# 1.4 25mm lens.

Colour image				Monochromatic image			
Predicted resolution		Measured resolution		Predicted resolution		Measured resolution	
9mm	11mm	9mm	11mm	9mm	11mm	9mm	11mm
7.7mm	9.4mm	9mm	11mm	15.4mm	18.8mm	16mm	20mm

From the images of the eye pattern and the table above it can be concluded that both the monochromatic and colour version of the MVC will be able to provide sufficient resolving power to adequately image 1cm and larger froth bubbles from distances ranging between 1- to 2m. Practically, the specific resolving power required to produce suitable images of froth will dependant on the type of image processing implement. This implies that depending on the image processing the correct MVC and lens setup will have to be chosen. If a higher resolving capability is required from the MVC a longer focal length lens can be used with the disadvantage of a reduced imaged area and higher lens cost.

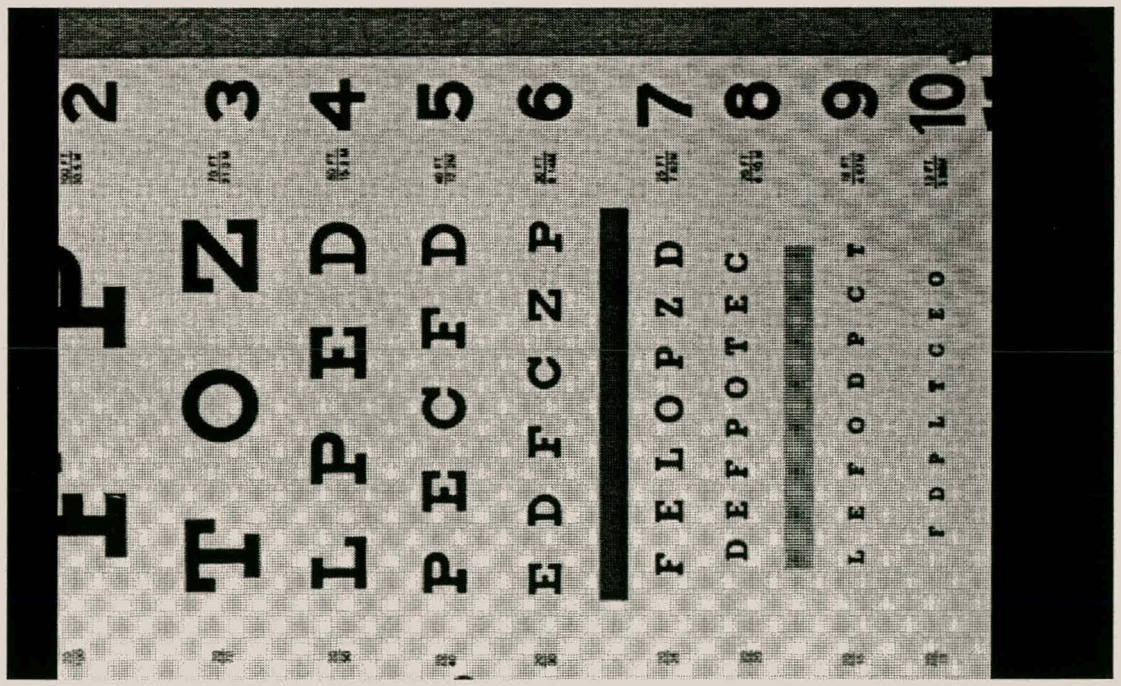

Figure 4.1 Monochromatic image of the horizontally placed eye pattern taken from a 2m distance.

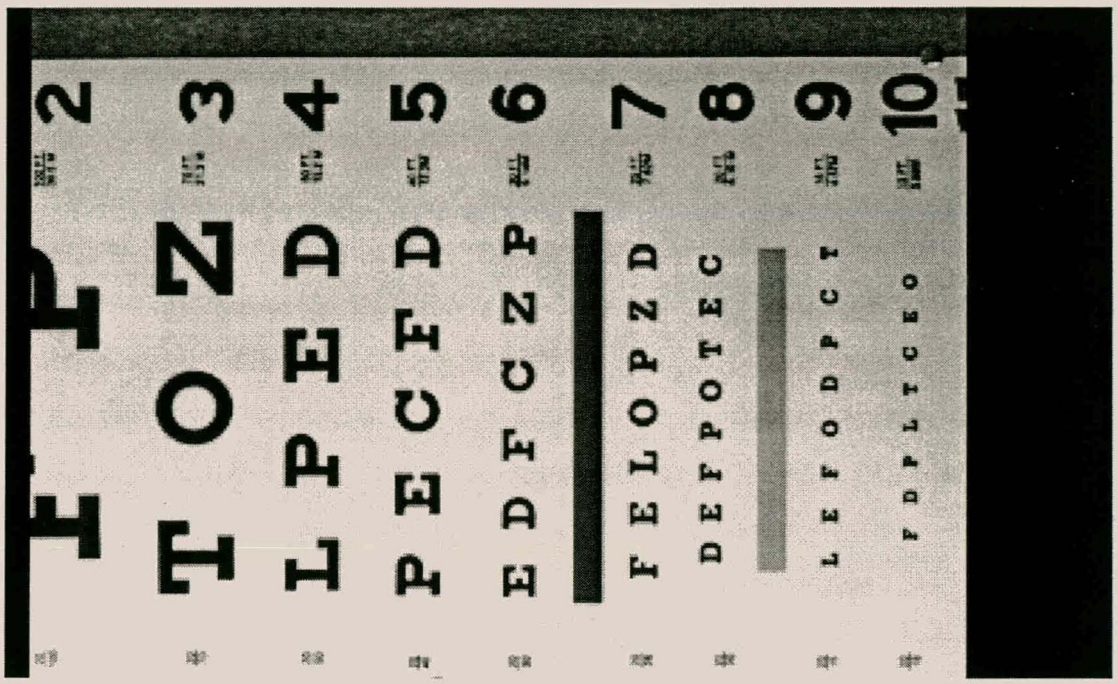

Figure 4.2 Monochromatic representation of the colour image of the horizontally placed eye pattern taken from a 2m distance.

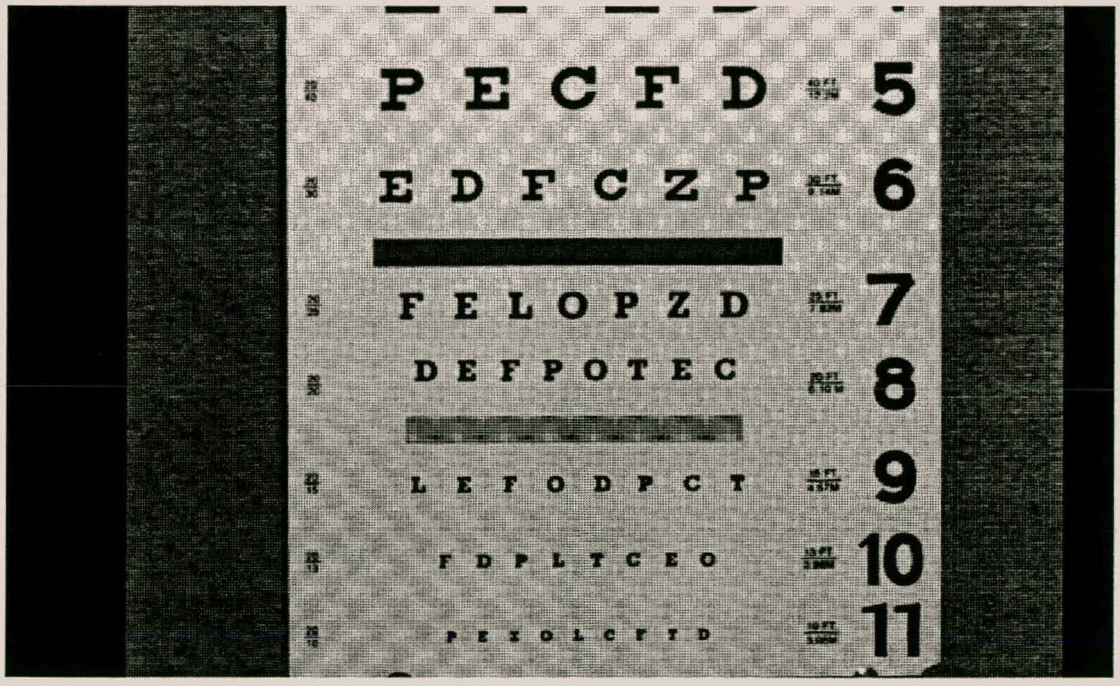

Figure 4.3 Monochromatic image of the vertically placed eye pattern taken from a 2m distance.

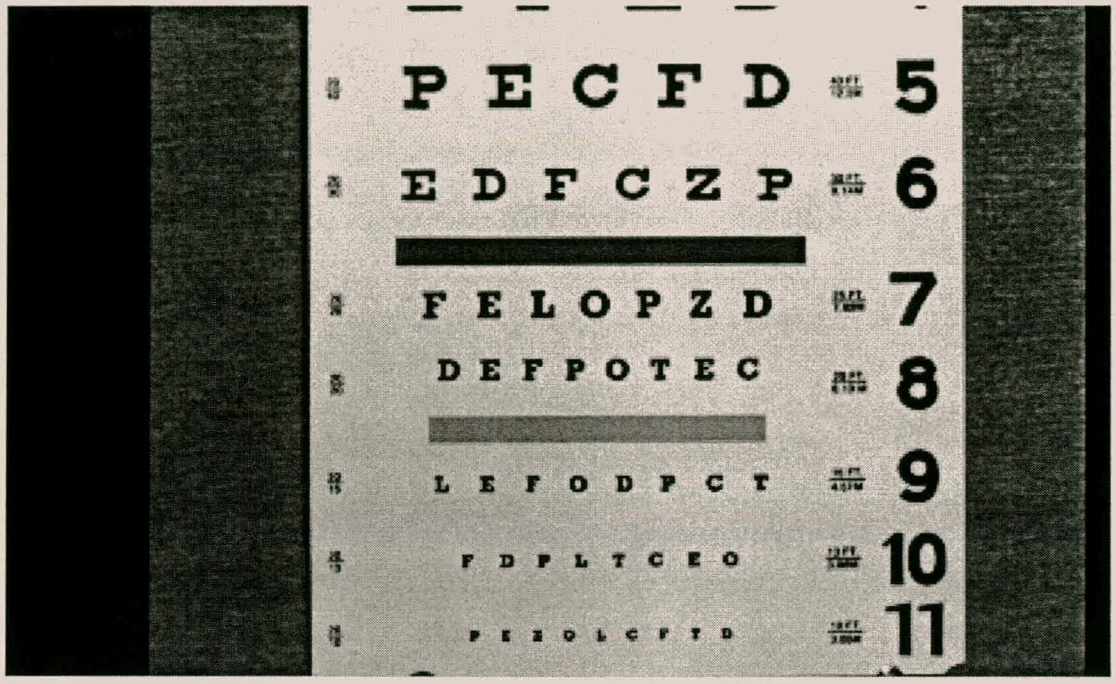

Figure 4.4 Monochromatic representation of the colour image of the vertically placed eye pattern taken from a 2m distance

4.3 Evaluation of the Image Capture Rate of the MVC.

The MVC as implemented for this thesis captures images at a constant rate of approximately 20 frames/s. The frame rate was chosen to ensure that the bursting of bubbles and the translation of the surface froth will have a minimal effect on the image quality, while trying the keep the image data rate from the MVC to a minimum. The same MVC and lens setup as in the previous paragraph is used, and all images shown are the monochromatic equivalent of the RGB colour images from the MVC.

This section will show that the value of 20 frames/s alone may not be sufficient to image highly mobile surface froth with minimal distortion; and that depending on the imaging distance and the translation speed of the surface froth, the integration time of the MVC may have to be shortened.

4.3.1 Evaluation.

In order to evaluate whether the 20 frames/s frame rate of the MVC will be sufficient to image surface froth moving at speeds in the order of centimetres per second, images of a test pattern rotating at a constant angular speed of 30°/s are captured. Figure 4.5 shows the test pattern at rest, when imaged from a distance of 2m. The test pattern contains seven 1cm in diameter circles spaced 2cm apart and three lines with a width of 1mm, 2mm and 4mm respectively. This high contrast pattern was chosen to ensure that the smear produced by the rotating image would be clearly visible. The aim of the test pattern is to determine how each circle is smeared in relation to one another, and to evaluate what degree of image distortion will be expected. Similarly, the lines are used to evaluate the smear on objects with higher spatial frequency detail.

Figure 4.6 shows an image of the same test pattern rotating at a rate of 30°/s, while Figures 4.7 and 4.8 show the same test pattern imaged from a distance of 1m. For a fixed image rate of approximately 20 frames/s and using a F# 1.4 25mm focal length lens, the MVC will have an expected smear of approximately 1.7 pixels when imaging an object moving at 1cm/s from a distance of 1m. At an imaging distance of 2m, the smear will be approximately 0.87 pixels.

Paragraph 2.4 states that the translation speed of surface froth can range from static to several centimetres per second. Thus, the amount of smear created by the MVC will be directly proportional to the translation speed of the surface froth. An important fact to note is that the percentage smear created for an object with a fixed translation speed remains constant for a change in the imaging distance. This is due to the fact that the lens determines the resolving power of the MVC, and causes the resolving power of the MVC to decrease linearly for an increase of imaging distance.

With this knowledge in mind, we can now evaluate the smear from the MVC. From Figures 4.6 and 4.7 it can be seen that the smear for the individual circles visually increases from the centre circle to the outer circle. The centre circle rotates around its

own origin with no visible distortion; while the outer circle shows the most distortion. Since the centre point of the outer circle is placed 12cm away from the origin of the centre circle, the centre of the outer circle will be translated at a speed of approximately 6.3cm/s. Under these conditions, an average smear of approximately 5.6 and 11 pixels is expected for Figures 4.6 and 4.7 respectively. Since Figures 4.6 and 4.7 are the monochromatic versions of the RGB colour images, the visible smear will only be seen as 2.8 and 5.5 pixels.[1] Practical measurements from Figures 4.6 and 4.7 show the smear to actually be 3 and 5 pixels respectively.

Similarly, from Figures 4.6 and 4.8 it can be seen that the three lines progressively smear more towards the edge of the test page. An important fact to note is that the 1mm line seems to be more visually smeared than the thicker lines. In reality, all three lines experience the same amount of smear. The only difference is that the percentage smear for the 1mm line is twice the smear for the 2mm line and four times more than the 4mm line.

From these images it can be deduced that an object with a higher spatial frequency will appear more visually distorted (smeared) than an object with a lower spatial frequency moving, at exactly the same speed. From Figures 4.6 to 4.8, it can be concluded that the MVC will be able to suitably image slow moving surface froths (e.g. < 6cm/s), while images of highly mobile surface froths may be severely distorted. Unfortunately, this places a limitation on the ability of the MVC to monitor various types of flotation plants. The following paragraph explains how the current MVC can be modified to ensure that the MVC can be used to image highly mobile froths.

[1] Note: Four pixels from the MVC's CCD sensor are required to create a single colour pixel.

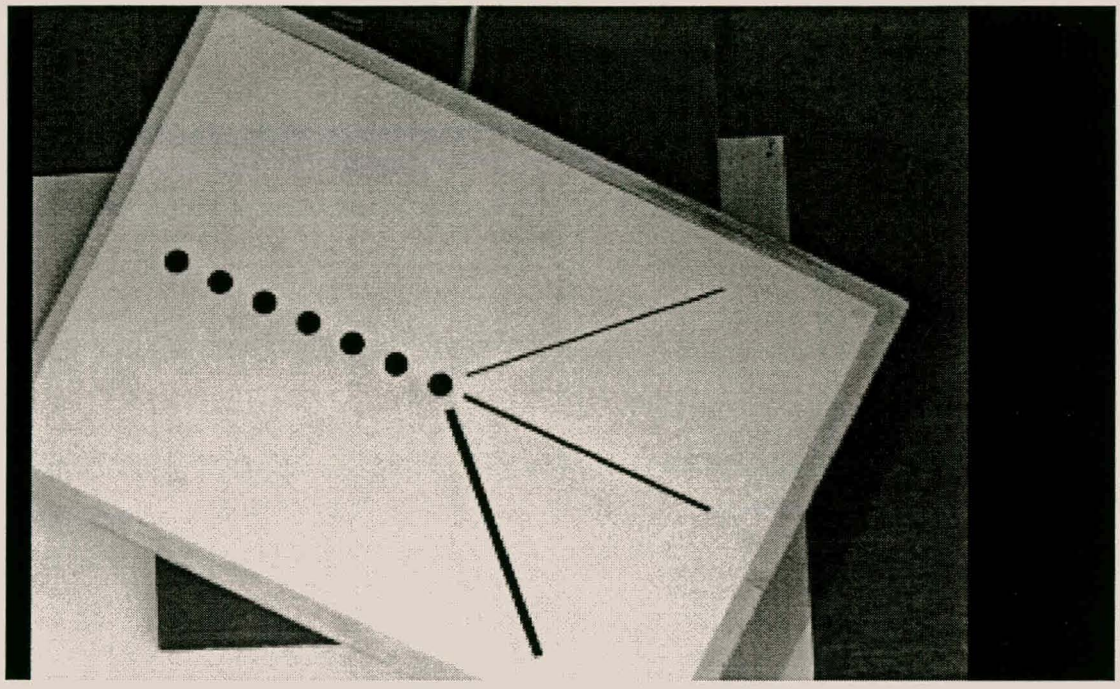

Figure 4.5 Image of MVC smear test pattern from a distance of 2m.

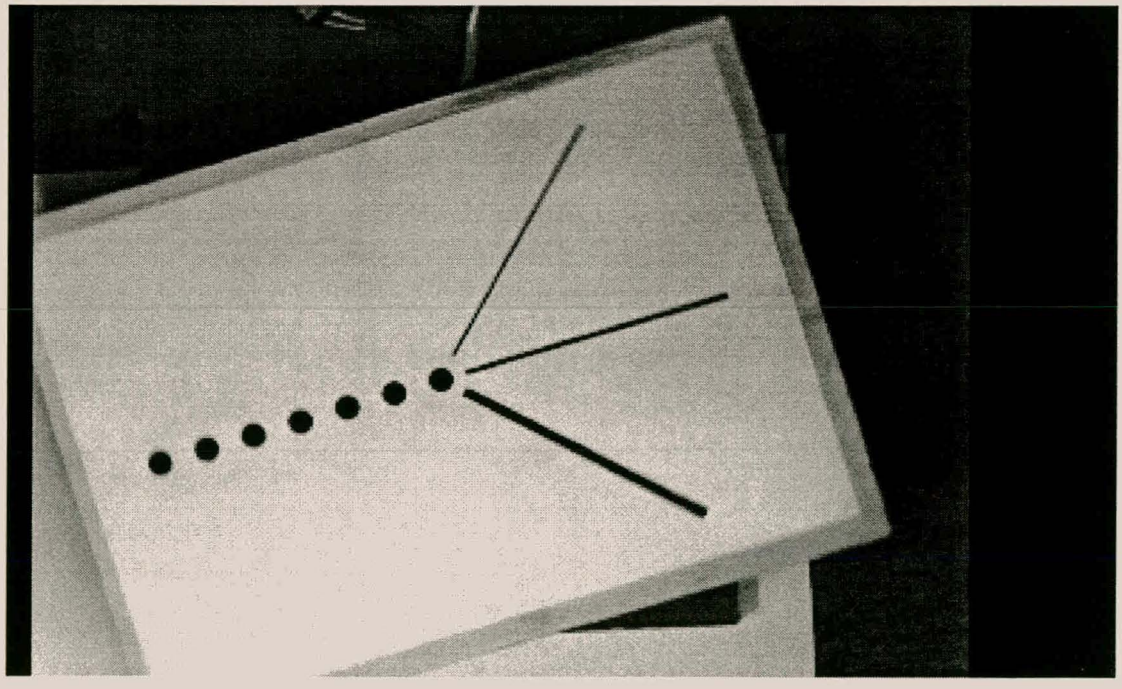

Figure 4.6 Image of MVC smear test pattern rotating at a rate of 30°/s.

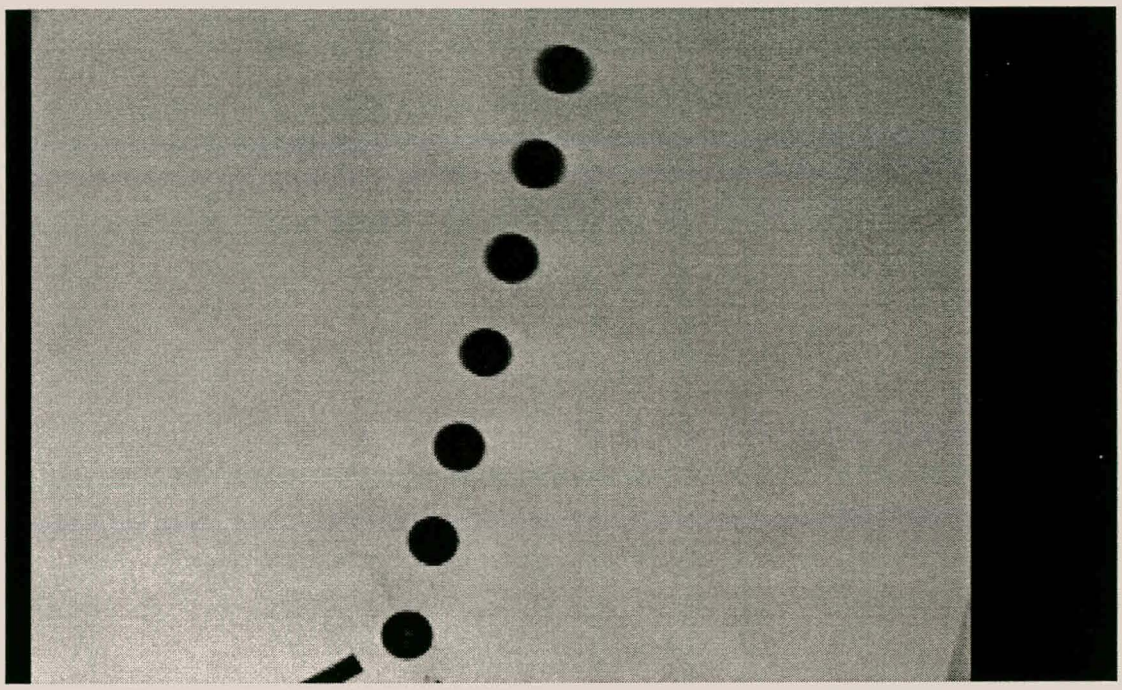

Figure 4.7 Upper half of MVC smear test pattern imaged from a distance of 1 m, while rotating at an angular rate of 30°/s.

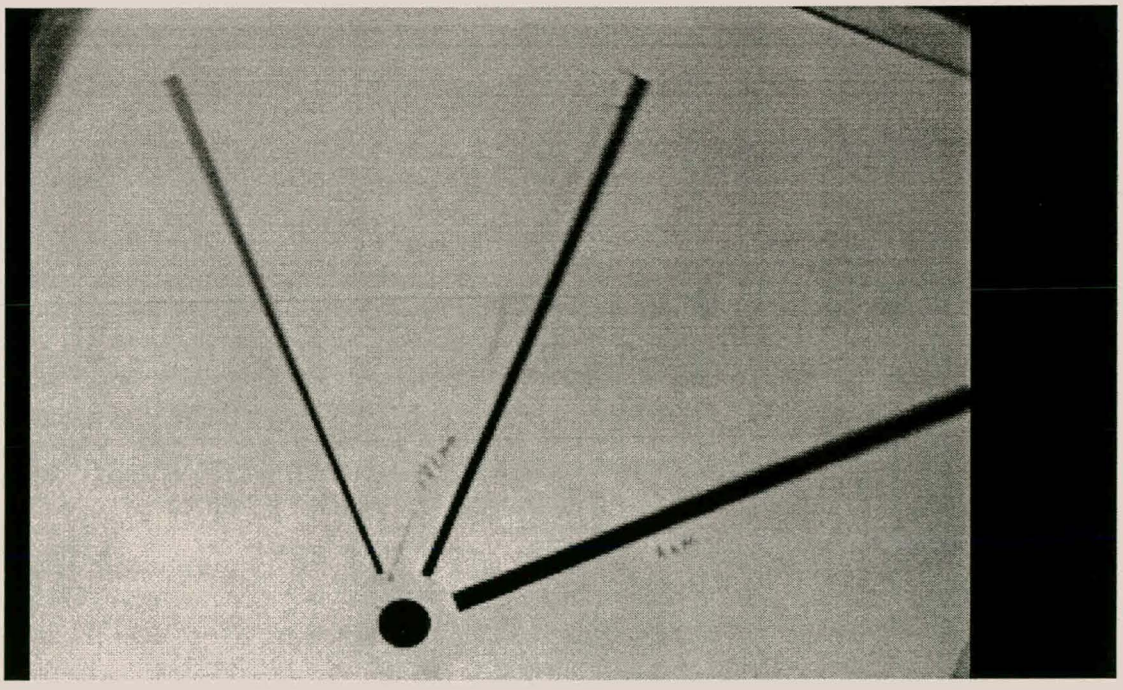

Figure 4.8 Lower half of MVC smear test pattern imaged from a distance of 1m, while rotating at an angular rate of 30°/s.

4.3.2 Enhancing the Imaging Capability of the MVC.

To ensure that smear will not influence the imaging performance of the MVC, the exposure time of the CCD sensor of the MVC must be shortened. This will ensure that the translation speed of the surface froth being imaged will cause minimal or no distortion (smear) in the captured images. For example, to ensure that surface froth moving at a translation speed of 6cm/s, while been imaged from a distance of 1m using a F# 1.4 25mm focal length lens, will cause no more than 1 pixel smear it is necessary to change the current exposure time of the MVC from 1/20s to approximately 1/208s or faster. This method of controlling the exposure time of a CCD sensor is commonly known as shuttering. Fortunately, both the monochromatic and colour CCD sensor of the MVC support electronically shuttering, which is controlled by the Timing Generator of the MVC. Unfortunately, for the current implementation of the MVC, the electronic shuttering function of the Timing Generator is disabled in hardware (paragraph 3.3.1).
The Electronic shutter of the Timing Generator has the following shutter modes, of which only the first two shutter modes will be required for flotation:

1. Electronic shutter off: Exposure time is 1/30s (1/20)[2] (Current setting of the MVC.).
2. High- speed electronic shutter: Exposure time is shorter than 1/30s (1/20).
3. Low-speed electronic shutter: Exposure time is longer than 1/30s (1/20).
4. Flickerless: Exposure time is 1/50s (1/33).

The Electronic shutter of the Timing Generator can be accessed in serial or parallel. The primary difference between the two methods is that the serial method allows the exposure time to be adjusted in steps of one horizontal scan period (approximately 100µs for a 15.98MHz crystal), while the parallel method only supports a fixed amount of high-speed shutter settings.

[2] The exposure times in brackets are when the Timing generator is clocked with a 15.98MHz crystal.

Table 4.3 High-speed shutter settings for the Timing Generator using the parallel method.

Mode	Exposure time
Electronic shutter off	1/ 30s (1/20)
Flickerless	1/50s (1/33)
High-speed shutter	1/60s (1/39)
	1/125s (1/81)
	1/250s (1/163)
	1/500s (1/325)
	1/1000s (1/651)
	1/2000s (1/1302)
	1/4000s (1/2604)
	1/10000s (1/6510)

Note : The exposure times in brackets are when the Timing generator is clocked with a 15.98MHz crystal.

Table 4.3 summarises the high-speed shutter setting for the parallel method. Thus, from the discussion above it can be deduced that both methods for setting the Electronic shutter of the Timing Generator are suitable. Since the serial method can set the exposure time to a finer resolution, it is suggested that this method is used to adjust the exposure of the Timing generator.

Figure 4.9 shows a basic block diagram of the proposed modification to allow the adjustment of the integration time of the MVC. With the exception of the shaded parts of Figure 4.9, the architecture for the MVC is the same as that described in Chapter 3. The only difference is the addition of a microcontroller to interface with the Electronic shutter control port of the Timing Generator (paragraph 3.3). This will set the shutter speed of the CCD sensor, and instruct the Timing Generator when to integrate and clock an image from the CCD sensor (External Triggering). The additional requirement to externally trigger the Timing Generator when to integrate and clock an image from the CCD sensor comes from enabling the electronic shutter function of the Timing Generator. The Timing Generator data sheets (SONY, 1997a:1685) provide more detail regarding this requirement.

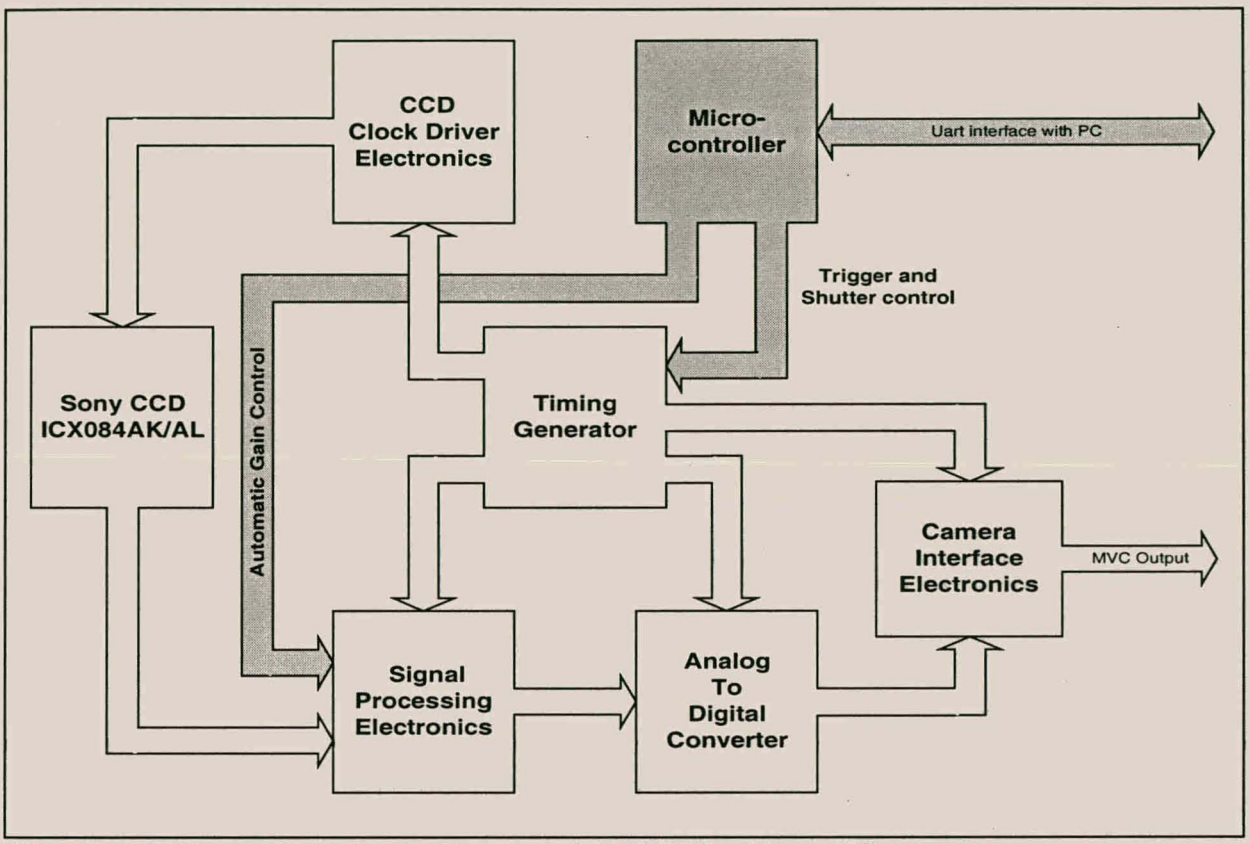

Figure 4.9 New general block diagram of the MVC with modification.

From Figure 4.9 it can also be seen that a path from the microcontroller to the Signal Processing Electronics (Correlated Double Sampler and Head Amplifier) has been added. Since the integration time of the MVC will now be adjusted according to the speed of the surface froth, it will be necessary to adjust the signal gain of the Head Amplifier to compensate for the loss of illumination as the integration time of the CCD sensor is shortened. This modification will necessitate the replacement of the two variable potentiometers, used to the set the gain of the Head Amplifier (CXD1690Q, paragraph 3.6.2.4), with digital potentiometers that can be programmed through the microcontroller.

An important fact to note is that this modification will not require the user to continuously adjust the integration time and gain of the MVC. The integration time is initially adjusted to the minimum value needed for the maximum expected froth speed.

The gain is then adjusted accordingly. As a result, the autonomous functionality of the MVC is maintained.

With the Electronic shutter control of the Timing Generator enabled, the frame rate of the MVC will now be lower than the current setting of 20 frames/s. The following equation defines the new relation between the frame period and the shutter speed :

$$Frame\_period = 33.367\times10^{-3}\cdot\left(\frac{f_{crystal}}{24.5454\times10^{6}}\right)+shutter\_period \qquad (4.2)$$

where $f_{crystal}$ is the clock frequency of the crystal used to drive the Timing Generator.

Fortunately, the change in the frame rate from the MVC will have no impact on the MVCI, since the data rate of the images is unchanged by Electronic shuttering. This is due to the fact that the data rate is a function of $f_{crystal}$ as shown by the first part of equation 4.2.

4.4 Evaluating the Dynamic Range of the MVC.

The Dynamic Range of a CCD sensor is a function of the sensor's support electronics, optics, and the illumination used. As a result, the DR is typically broken into an Electrical DR (EDR) and an Optical DR (ODR) measurement and is defined by the following equations (DALSA, 1997c:304) :

$$EDR = 20\cdot\log\left(\frac{V_{A/D}}{Vnoise}\right) \qquad (4.3)$$

where $V_{A/D}$ is the maximum input voltage to the A/D converter of the MVC and *Vnoise* is the total rms. random noise voltage of the MVC, when images are captured in the dark.

$$ODR = 20 \cdot \log\left(\frac{SEE}{NEE}\right) \qquad (4.4)$$

where SEE (Saturation Equivalent Exposure) is the upper bound of the usable input range of the external light source. This is applied within a narrow band, centred at the wavelength of peak responsivity, causing the CCD sensor of the MVC to saturate. NEE (Noise Equivalent Exposure) is the exposure (input light signal), applied within the same narrow band centred at the wavelength of peak responsivity. This is done to produce signal levels equivalent to the rms. noise (*Vnoise*) of the MVC.

For the DR evaluation of the MVC, only the EDR will be considered, since the ODR is a function of the exposure time, specific optics and source of illumination used for the MVC.

Paragraph 2.11.4 mentioned that a theoretical maximum DR of approximately 56dB can be expected for both the monochromatic and colour CCD sensor of the MVC. The EDR of the 8-bit A/D converter will be in the range of approximately 48dB to 60dB. Therefore, the overall EDR of the MVC should fall within the range of 48dB to 56dB. To calculate the random noise (*Vnoise*) of the MVC, successive images are captured with the MVC in total darkness. The difference between the images is considered to be noise, and is quantified by the following equation (DALSA, 1997d:314):

$$Vnoise = \frac{STDDEV\left(IM_1 - IM_2\right)}{\sqrt{2}} \qquad (4.5)$$

were *STDDEV* is the two-dimensional standard deviation function; IM_1 and IM_2 are two successive images from the MVC; and *Vnoise* is the rms. random noise of the MVC in A/D units (ADU).

The results of this analysis are shown in Table 4.4 when five successive images were captured at a rate of approximately 20 frames/s with the MVC in total darkness. The EDR

was calculated for each individual colour component of the MVC, and for the overall image where each pixel is considered to be monochromatic.

Table 4.4 Results of the EDR measurement for the MVC.

$IM_{T1} - IM_{T2}$	R pixels	Gr pixels	Gb pixels	B pixels	All pixels
$Vnoise_{1-2}$	1.381 ADU	1.212 ADU	1.382 ADU	1.211 ADU	1.216 ADU
$Vnoise_{2-3}$	1.382 ADU	1.212 ADU	1.385 ADU	1.219 ADU	1.218 ADU
$Vnoise_{3-4}$	1.382 ADU	1.211 ADU	1.380 ADU	1.216 ADU	1.216 ADU
$Vnoise_{4-5}$	1.382 ADU	1.215 ADU	1.379 ADU	1.209 ADU	1.215 ADU
Ave. *Vnoise*	1.382 ADU	1.212 ADU	1.382 ADU	1.214 ADU	1.216 ADU
Ave. EDR	45.356dB	46.493dB	45.358dB	46.482dB	46.464dB

From Table 4.4 it can be seen that the EDR for the MVC is approximately 1.5dB lower than the minimum expected value of 48dB. In practice, the lower EDR value for the MVC will have no effect on the capability of the MVC to image and extract relevant froth features from the surface froth of a flotation cell, as long as the full EDR of the MVC is used. Since the external light source and MVC setup for industrial flotation are under the user's control, this condition places no constraint on the use of the MVC.

4.5 Conclusion.

In this chapter, the MVC was placed through three tests; to determine whether it will produce suitable images of dynamic surface froth.

The first test consisted of capturing images of a standard optometrist eye pattern, and evaluating the resolving power of the MVC for the text of the pattern. From this test, it was deduced that with the correct focal length compound TV lens; a high resolving power can be achieved with the MVC. In practice, the specific resolving power needed will depend on the image processing implemented to extract the relevant froth features. This will typically require a compromise between the imaged area and resolving power for the MVC.

The second test evaluated whether the MVC will be able to produce images of mobile froth with minimal distortion. For this test, images of similar objects moving at different speeds were captured. The distortion (smear) in each image was measured and compared to predicted values. From the test results, it was ascertained that depending on the mobility of the froth; the 20 frames/s image capture rate of the MVC alone may not be sufficient to image highly mobile froth (faster than 6cm/s), and that electronic shuttering and variable gain control will have to be implemented for the MVC.

The third and final test evaluated the EDR of the MVC. Images were captured while the MVC was in the dark, and the rms. noise of the images were evaluated. The test showed that the EDR of the MVC is approximately 46.5dB, which will be sufficient to image dynamic surface froth so long as the full EDR of the MVC is used. Examples of RGB colour images captured from the MVC can be found in Appendix H.

CHAPTER 5

EXTRACTION OF RELEVANT FROTH FEATURES.

In Chapter 1 it was shown that the structure of froth in Froth Flotation plants can be classified on the basis of textural features extracted from the images using digital image processing techniques. Through the application of these techniques, a relation between the average bubble size, colour, texture, stability, and mobility of the surface froth to the grade and recovery of the flotation process was identified. From the same results, the mobility (speed) and stability were identified to be the two most important froth features, followed by the average bubble size. In Moolman's study; methods for determining the stability, mobility and average bubble size of the surface froth were also formulated (Moolman, 1995a).

This chapter investigates the use of image cross-correlation as a useful tool to determine the mobility and stability of the surface froth, based on previous work done by Moolman. A discussion of the present methods used to characterise the mobility and stability of the surface froth, along with an overview of some of their advantages and disadvantages, is presented. This is followed by an analysis of image cross-correlation as a useful tool to measure the stability and mobility of the froth. This method is then evaluated using video footage and practical measurements of froth.

5.1 Measurement of Froth Mobility.

Using the assumption that the flotation froth can be viewed as a texture, various textural features (*SNE*, *LNE*, *NNU*, *SM*, etc) can be extracted from an image of the surface froth by using a Neighbouring Grey Level Dependence Matrix (NGLDM) method.

Moolman finds that by relating the mobility of the surface froth to the *SM* (second moment) textural parameter, an accurate measure of the froth speed can be obtained (Moolman, 1995a:132). Similarly, the *NNU* (number non-uniformity) and *SNE* (small number emphasis)

parameters were found to give a good qualitative measure of the average bubble size of the surface froth.

The NGLDM is a two-dimensional histogram $Q(r,s)$, where Q can be considered as an indication of the frequency counts of greyness variation of an image. The dimensions of the Q matrix are r x s where r represents the number of grey levels and s the number of possible neighbours to a pixel in an image. The Q matrix can be computed by counting the number of times that the difference between each pixel in an image and its neighbours at a certain inter-pixel distance d are equal to or less than a threshold value h.

The following histogram features can be determined from the NGLDM (Q) :

- Small Number Emphasis $SNE = \sum_r \sum_s \left[Q(r,s) / s^2 \right] / R$ (5.1)
- Large Number Emphasis $LNE = \sum_r \sum_s \left[s^2 \cdot Q(r,s) \right] / R$ (5.2)
- Number Non-uniformity $NNU = \sum_s \left[\sum_r Q(r,s) \right]^2 / R$ (5.3)
- Second Moment $SM = \sum_r \sum_s [Q(r,s)]^2 / R$ (5.4)

where R is a normalising factor $R = \sum_r \sum_s Q(r,s)$

The method used to measure the mobility of the surface froth using the *SM* parameter is based on the knowledge that when imaging a scene with a digital camera, the scene will be blurred or smeared as a linear function of the exposure time of the camera and the relative velocity between the scene and camera. As a result, images of froth with a high mobility will produce more blurred or smeared images when compared to images of stagnant or slow-moving froth. Since the *SM* parameter is a qualitative measure of the homogeneity of an image; the more an image is smeared or blurred, the more homogenous the image will appear.

The use of exposure control (to increasing the integration time) has the disadvantage that images used for the mobility measure cannot be used for stability measures, since the images are "smoothed" as a linear function of the exposure time. This disadvantage will become clearer when the stability measure is explained in the following paragraph. Another disadvantage in the use of thresholding. In practise, the threshold h is dependent on the illumination of the surface froth[1], which in turn is dependant on the external lighting, bubble structure and the amount of minerals present on the bubbles of the surface froth. Variations in the lighting condition or the amount of minerals on the bubble surface will cause a change in the intensity bias of the surface froth, causing an incorrect measure of the froth mobility. To minimise the sensitivity of h to a bias change in the grey level distribution of the images; histogram equalisation of the grey level distribution of the froth images is implemented. This in turn increases the processing requirements of the mobility algorithm. An additional disadvantage of using SM to measure the mobility of the froth, is that often variations of the surface froth thickness may cause images of the froth to become defocused. This in turn can lead to an incorrect measurement of the froth mobility. Further information regarding this method and its use for texture classification may be found in the work of Sun and Wee (1982).

5.2 Characterisation of Froth Stability.

The froth stability can be characterised by the statistical distribution of the difference between the grey level values of two consecutive images and can be defined by the following function :

$$S = \sum_{j=0}^{M-1}\sum_{i=0}^{N-1} f(i,j) \qquad (5.5)$$

where $f(i,j) = \begin{cases} 1 & if \quad |IM_1(i,j) - IM_2(i,j)| \geq h \\ 0 & if \quad |IM_1(i,j) - IM_2(i,j)| < h \end{cases}$

[1] Assuming the configuration and physical setup of the machine vision camera are unchanged.

and IM_1 and IM_2 are two successive grey images and h is the grey level threshold value.

The characterisation of the froth stability by equation 5.5 is based on the following argument: If the froth is excessively stable, the rate of bubble collapse is slow and fewer local changes between successive images should occur than in an unstable froth. For an unstable froth, the high rate of bubble collapse and formation of new bubbles will cause quick local changes between successive images. The value of S can therefore be considered as an indication of the number of local variations between successive frames, and hence, an indication of the instability of the froth.

Unfortunately, Moolman reports that there is no explicit stability measure with which S can be compared. At present only a qualitative indication is obtained by comparing S with visual observations and process parameters known to affect the stability of the froth (Moolman, 1995a:130).

An advantage of using equation 5.5 to characterise the froth stability is that it places a low requirement on the image processing system. Unfortunately, the same simplicity is also responsible for the problems and practical limitations experienced with this method.

It can be seen from equation 5.5 that S is based on the assumption that the mobility of the froth is very slow compared to the time between successive images; and that the threshold h is assumed to be constant over the period in which stability is measured.

Since the stability measure also uses thresholding, the measure will be sensitive to variations of the froth illumination as described in paragraph 5.1. As a result, histogram equalisation is again implemented to minimise the sensitivity of the stability measure to changes in illumination. A large disadvantage of S is its high sensitivity to the froth mobility. To make S invariant to the mobility, the frame rate between successive images has to be sufficiently higher than the mobility of the froth in order to ensure that the translation of the froth will not adversely influence the stability characterisation.

Depending on the physical MVC setup (CCD sensor size, optics, imaging distance) and the mobility of the froth, equation 5.5 can place an unnecessary restraint on the image processing system of the MVC to characterise the stability.

The most straightforward manner in which to illustrate this restraint is by means of an example: the assumption is made that a froth is moving at a typical rate of 1cm/s. A typical MVC setup for flotation is used, where images are captured through a 25mm F1.4 lens at a distance of 2m using the MVC described in this thesis. Assuming that a translation of no more than one image pixel is required to make S invariant to mobility will imply that successive images must be captured at a rate of approximately 17 frames/s. This example clearly shows that depending on the physical camera setup and flotation plant being imaged, the stability measure can require an image processing system to capture images at a relatively high rate.

5.3 Image Cross-Correlation as a Tool for Measuring the Mobility and Stability of Froth.

Paragraphs 5.1 and 5.2 show that the principle of the methods used to characterise the mobility and stability of the surface froth is based on directly or indirectly detecting variations between successive images of the froth. Through this reasoning it can be intuitively argued that by correlating the information content between two successive images, it should be possible to determine the mobility of the froth and also obtain a qualitative indication of the froth stability.

The concept behind the image cross-correlation method is simple: when two successive images of froth are cross-correlated, the position of the correlation peak will give an estimate for the mobility of the froth, while the amplitude of the peak in return can give an indication of the froth stability.

Since an image from a digital camera is a discrete two-dimensional spatial function, discrete two-dimensional correlation, as shown by equation 5.6, is used to implement the discrete image cross-correlation (DICC) function.

$$p(x,y)=\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\left\{\left[IM_2(i,j)-\overline{IM_2}\right]\cdot\left[IM_1(x+i,y+j)-\overline{IM_1}\right]\right\} \qquad (5.6)$$

where M and N are the vertical and horizontal resolutions of an image respectively. IM_1 and IM_2 are two successive grey level images, while $\overline{IM_1}$ and $\overline{IM_2}$ are the average grey level values for the two images.

The subtraction of the average grey level value from each image ensures that on average information not present in both images will not contribute to the correlation process. As a result, only information contained in the image overlap area between successive images, will add to the cross-correlation process. The removal of the average grey level also has the added advantage of making equation 5.6 invariant to changes in the average grey level intensity of successive images.

To illustrate the use of discrete image cross-correlation to determine the mobility of a froth, refer to Figure 5.1. This figure shows a 248 x 248 pixel grey scale image, taken with a VHS video camera, of surface froth with a relatively high stability and low mobility. Let us assume Figure 5.1 represents a flotation cell with a surface froth of unknown mobility and the solid- and dashed squares (101 x 101 pixels) represent two successive images of the froth. For this example, the centre of the dashed square is translated +40 pixels in both the horizontal- and vertical direction away from the centre of the solid square. Let us now apply the DICC of equation 5.6 on the two images, with IM_1 the solid square and IM_2 the dashed square.

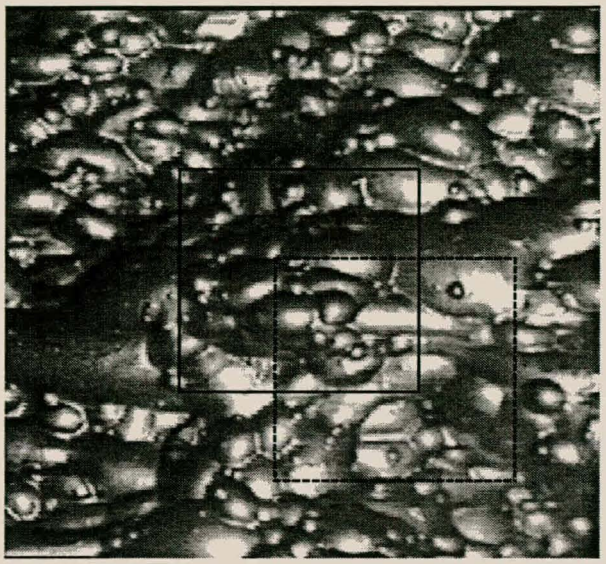

Figure 5.1 Image of surface froth from an industrial flotation cell.

Figure 5.2 shows the results of the DICC process seen from both the horizontal and vertical direction. The peak of the cross-correlation has shifted by a value of +40 pixels in both the horizontal and vertical axis away from the cross-correlation centre (101, 101). This fits the translation between the two squares exactly.

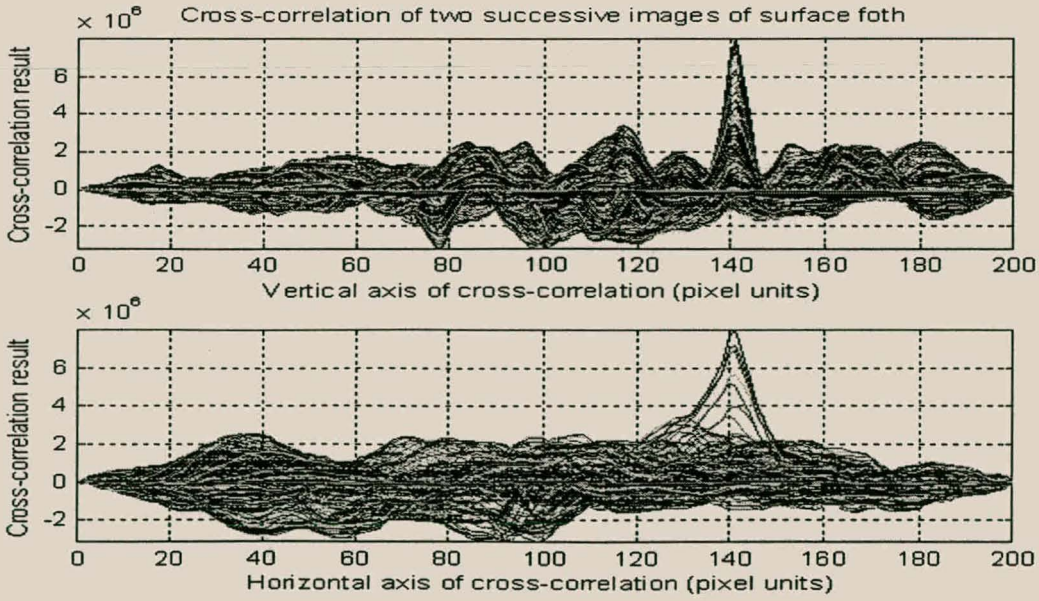

Figure 5.2 Image cross-correlation example result.

The mobility of the surface froth can be determined from the displacement of the cross-correlation peak by realising that the displacement is a function of the physical camera setup (type of optics used, imaging distance, etc.) and the time between successive images. The following equation describes this relationship:

$$Horizontal\ mobility = \frac{Horizontal\ pixel\ size}{M \cdot \Delta T} \cdot \Delta column \tag{5.7}$$

$$Vertical\ mobility = \frac{Vertical\ pixel\ size}{M \cdot \Delta T} \cdot \Delta row \tag{5.8}$$

$$|Mobility| = \sqrt{(Horizontal\ mobility)^2 + (Vertical\ mobility)^2} \tag{5.9}$$

where *Horizontal-* and *vertical pixel size* are the physical size for a pixel of the digital camera. Δrow and $\Delta column$ are the vertical and horizontal displacement of the cross-correlation peak from the cross-correlation origin in units of pixels. ΔT is the time between the two successive images and M is the magnification factor for the camera setup.

Thus, before the DICC function can be used to measure the mobility of froth in units of m/s, calibration of the camera setup is required. In practice, the success of using the DICC function of equation 5.6 to determine the mobility of froth is a function of the information contained in the percentage area overlap between successive images. The percentage area overlap itself is a function of the froth mobility and the time between successive images and can be described by the following equation:

$$\%\ Area\ overlap = \frac{\left(IM_{row\ size} - |\Delta row|\right) \cdot \left(IM_{column\ size} - |\Delta column|\right)}{IM_{row\ size} \cdot IM_{column\ size}} \tag{5.10}$$

where $IM_{row\ size}$ and $IM_{column\ size}$ are the vertical and horizontal resolutions of an image respectively. Δrow and $\Delta column$ are the vertical and horizontal displacement of the cross-correlation peak from the cross-correlation origin.

Due to the dependence of image cross-correlation on the percentage image area overlap and the information contained in the overlap area, the time difference (ΔT) between successive images must be optimised (maximised) for the specific flotation circuit to be measured. This implies that, depending on the specific flotation process being imaged, the image capture rate to the image processing system (processor) can be minimised.

5.4 Evaluation of Characterising Mobility and Stability with DICC.

To practically evaluate the possibility of using DICC to characterise the mobility and stability of froth, video images of batch and industrial flotation cells were used.

5.4.1 Stability.

To evaluate the stability, a small batch flotation cell (3-litre open top Leeds cell) was imaged, with the proposed MVC described in Chapter 3.

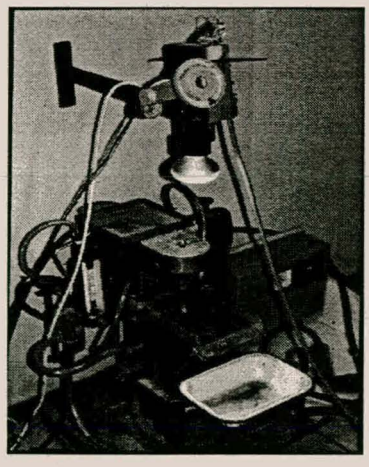

Figure 5.3 Batch flotation cell and camera setup.

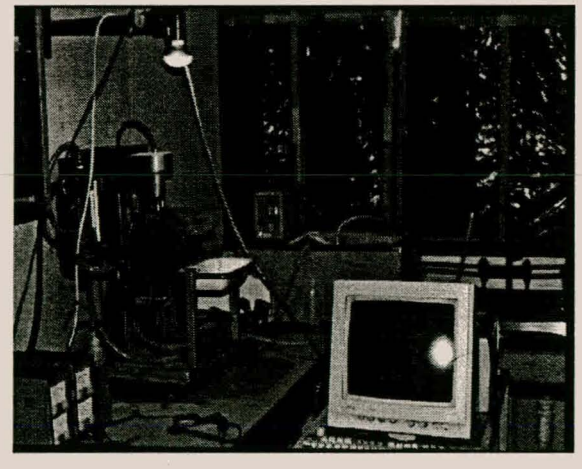

Figure 5.4 Batch flotation cell, Personal computer, and camera setup.

Figures 5.3 and 5.4 show the physical camera and batch flotation cell setup. Sand was used as the mineral for the flotation process, since it is the simplest type of flotation, and is commonly used in industry to obtain high quality silicon oxide for the production of silicon.

The conditions of the MVC setup and flotation cell were as follows: the camera was mounted approximately 50cm above the flotation cell. The camera was running at a frame rate of approximately 20 frames/s, while every hundredth frame was captured with the MVCI card[2]. The impeller of the cell was running at 1200rpm, while air was added at a rate of 4 litres/minute.

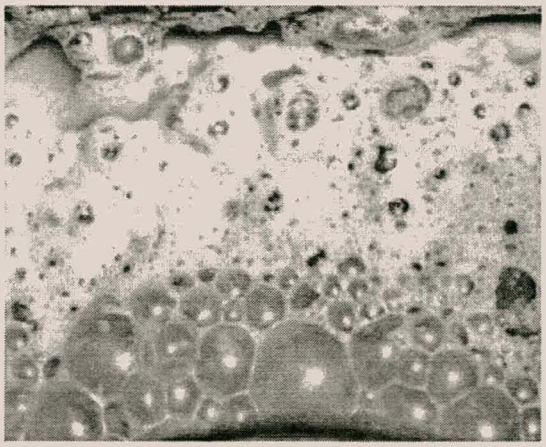

Figure 5.5 Image of froth at 200 seconds.

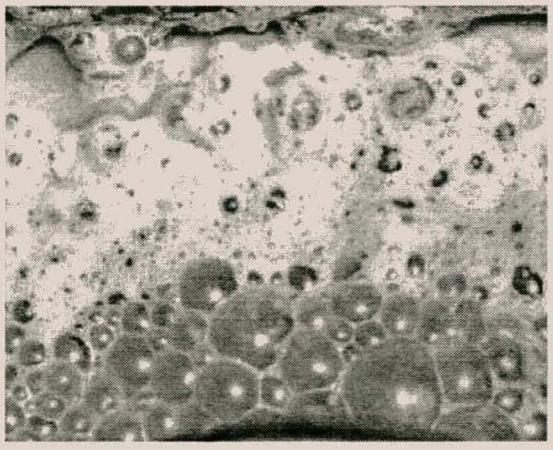

Figure 5.6 Image of froth at 205 seconds.

Figures 5.5 and 5.6 show two successive images of the surface froth, 200 seconds after the start of image capture. From the two images, it can be seen that a sample rate of 1 frame per 5 seconds is sufficient to capture the dynamic response of the froth, since no large local changes have occurred between the images. The large white area without bubbles is due to the tendency of this type of flotation to produce a "sticky" froth. Once a bubble bursts near the edge of the cell, the froth leaves a residue which prevents the further buildup of bubbles.

[2] Refer back to Chapter 3 for information regarding the MVCI card.

In the case of industrial flotation, this reaction will not occur, since the froth continously flows over the edge of the cell.[3]

A major problem with characterising the stability of froth, is that there is no measure against which to compare it. Therefore, to validly evaluate the stability measure, the results of the DICC process must be compared to visual observations and process parameters known to affect the stability of the froth. To do this, the flotation process was brought to a stable state at which time images were captured with the MVC. After approximately 250 seconds, the aeration flow of the flotation cell was stopped. This causes the formation of bubbles to stop, and as a result, the total loss of flotation.

To limit noise in the image cross-correlation result for stability, only grey level images were used; the reason for this being that the external light source did not equally stimulate the RGB components of the camera. By evaluating the RGB components of the images it was found that the green band of the camera was most stimulated by the external elimination, and as a result was used as grey level images.

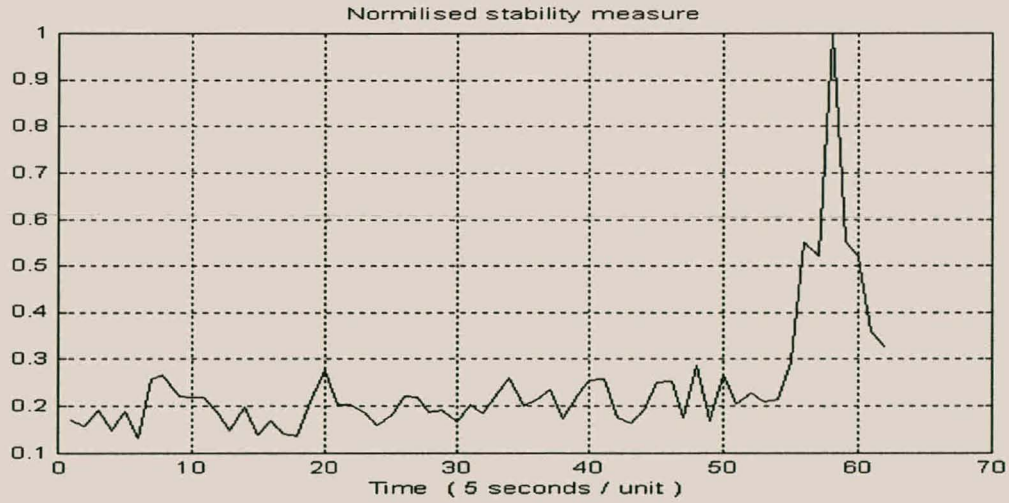

Figure 5.7 Measure of stability for the flotation process.

[3] In order to avoid the fringing of the froth affecting the stability measure, only the lower third (55 rows x 248 columns) of the image area is used.

Figure 5.7 shows the normalised result of the stability measure obtained by applying the DICC function (equation 5.6) on the green pixels of the images captured from the batch flotation cell. It can be seen from this figure that the stability measure has a low frequency variation around 0.2, indicating that the froth is in a stable state. This low frequency variation in the stability is caused by local changes in the froth between successive images by the formation and bursting of bubbles. At 250 seconds from the start of image capture, a large deviation is detected in the stability of the froth, which was caused by the halting of aeration to the process. From this result, it can be seen that the image cross-correlation stability measure was able to detect a large change in the stability of the froth.

To further illustrate the stability measure, images of an unknown batch flotation cell and flotation process were extracted from a VHS videotape. A 20 second grey level image sequence, at 2 frames/s, was captured. The sizes of the images were set to 240 rows x 320 columns.

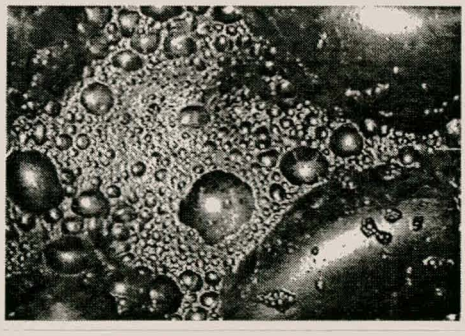

Figure 5.8 Image of flotation froth at 13 seconds.

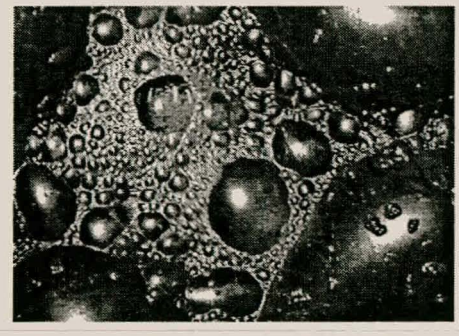

Figure 5.9 Image of flotation froth at 13.5 seconds.

Figures 5.8 and 5.9 show two successive images of the flotation process. From the two images it can be seen that a frame rate of 2 frames/s is sufficient to capture the dynamic response of the froth. As previously, the froth is stationary with no mobility.

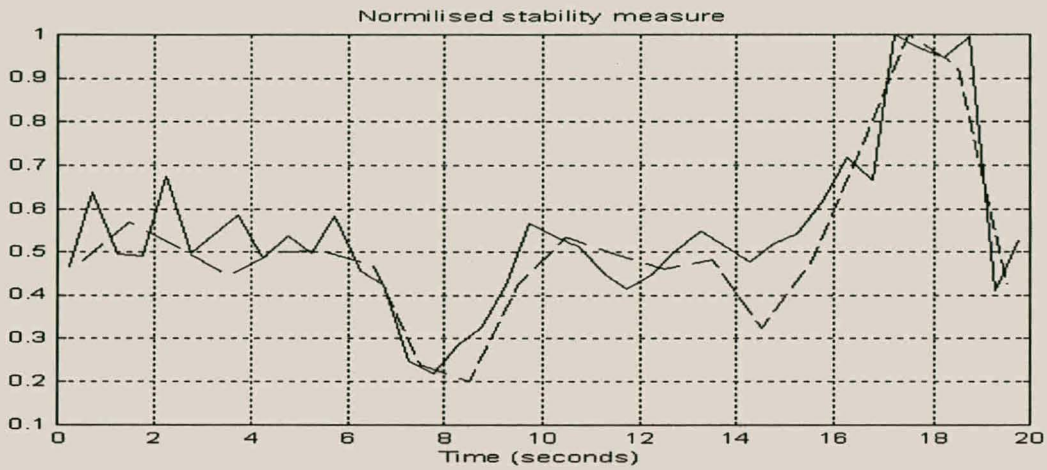

Figure 5.10 Measure of stability for the flotation process.

Figure 5.10 shows the normalised response of the stability measure for the 20 second image sequence of the flotation process using image capture rates of 2 frames/s (solid line) and 1 frame/s (dashed line). From this figure it can be seen that both image rates give a very similar stability measure for the flotation process, which indicates that for this flotation process an image capture rate of 1 frame/s is also sufficient to measure the stability. Two distinct deviations can be seen in both plots, one between 6 and 10 seconds and the second between 16 and 20 seconds. The first deviation is caused by the bursting of almost all the large bubbles of the surface froth and the gradual rebuild-up of new bubbles, while the second deviation is caused by the excessive build-up of large bubbles and the eventual bursting of these bubbles. Unfortunately, these deviations cannot be related to parameters of the flotation process, since no information is available on the process. The important fact to note is that the stability measure was able to detect these transients in the stability of the froth.

In the case of industrial floatation, the froth will be moving at a slow varying average mobility. In paragraph 5.3 it was noted that the stability measure is a function of the percentage image overlap, which in turn is a function of the froth mobility and the image capture rate of the MVC. Thus, for a fixed image capture rate, the stability measure will only be affected by the dynamics of the froth bubbles and changes in the mobility of the froth. For industrial flotation, the average speed of the froth does not change quickly as a function of time, since the time constants of industrial flotation plants range from tens of minutes to

hours. This implies that the stability measure should still give a relative indication of the overall stability for a mobile froth, but will more importantly still detect quick and large changes in its overall stability.

5.4.2 Mobility.

To evaluate the use of the DICC as a practical tool to determine the mobility of froth, images were extracted from a VHS videotape of an industrial flotation cell. A 30 second grey level image sequence was extracted from the video, at a rate of 4 frames/s. The sizes of the images were again set to 240 rows x 320 columns.

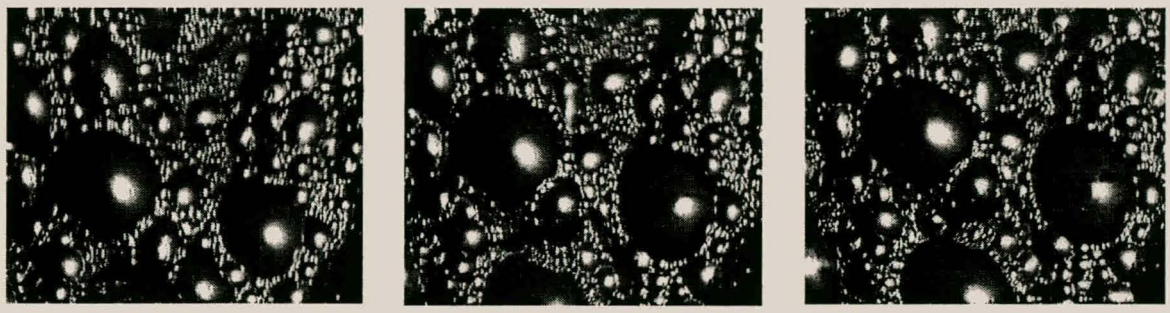

Figure 5.11 Three successive images of froth from an industrial flotation cell.

Figure 5.11 shows three typical successive frames of the image sequence. From the three images (left to right), the gradual upward motion of the froth is clearly noticeable. It is also clear that an image capture rate of 4 frames/s is sufficient to capture the dynamics of the froth.

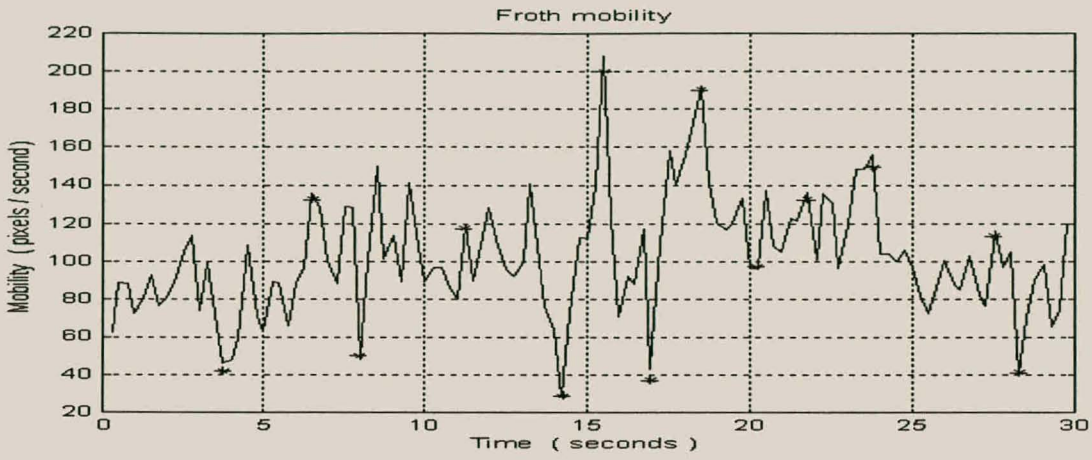

Figure 5.12 Mobility measure of froth through the use of DICC.

Figure 5.12 shows the result of measuring the mobility of the surface froth with DICC. Unfortunately, no reference data is available to compare the mobility of Figure 5.12 with the actual mobility of the froth. The reason for this it that very little or no information is available about the flotation process and the VHS camera setup used to film the process. To overcome this problem, the results of the DICC were compared with estimates of the froth mobility by visually looking at the displacement of the bubbles between successive frames. Five well-spread bubbles were selected from the images, and the displacement of the bright light reflections on top of the bubbles was used as a reference point. The result of this comparison is indicated by * for 13 points in Figure 5.12. In accordance with the results it can be seen that the crude mobility estimate and the DICC mobility measure compare fairly well, which indicates the DICC function of equation 5.6 was able to measure the mobility of the froth. The high frequency variation in the mobility measure can be attribute to local variations occurring in the surface froth, as old bubbles burst and new bubbles are formed. Furthermore, since no information is available about the flotation process and VHS camera setup, no attempt was made to evaluate the stability of the froth as done in Paragraph 5.4.1.

To further illustrate the success of the mobility measure, the mobility of the froth was again calculated using only 2 frames/s data from the same image sequence as used in Figure 5.12. The results of this analysis were then compared to an estimate of the expected mobility using the mobility results of Figure 5.12. Figure 5.13 shows the results of this analysis.

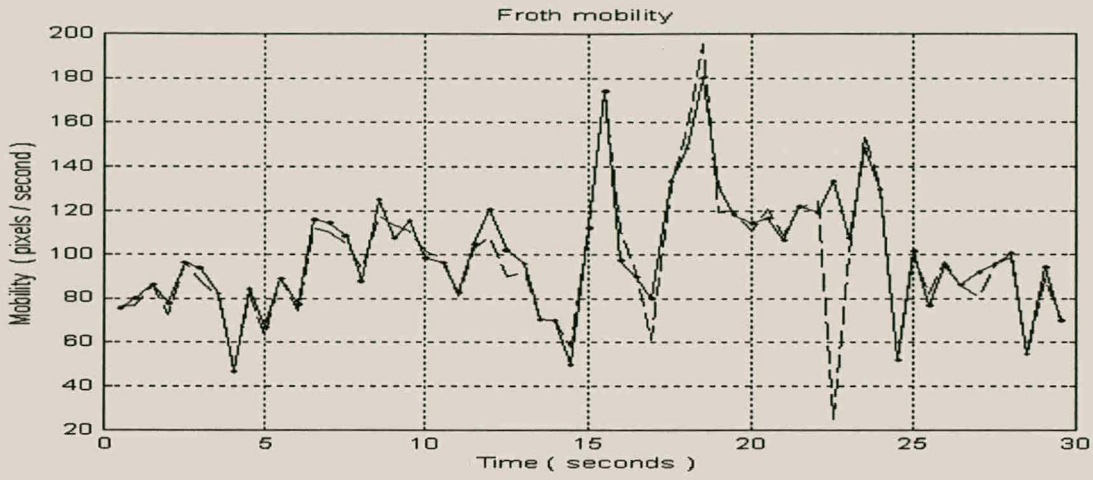

Figure 5.13 Comparison of the mobility measure between the 4 frames- and 2 frames / second images.

From Figure 5.13 is can be deduced that the mobility measure between the 2 frames/s (dashed line) and the 4 frames/s (solid dotted line) estimate fit closely. This indicates that both image capture rates were able to estimate the mobility of the froth. Only at 22.5 seconds does the 2 frame/s mobility measure drastically deviate from the 4 frames/s expected value. The reason for this is that a fast local change occurred in the froth structure during this period, which was not detected by the 2 frames/s image sequence. An interesting fact to note is that for the 4 frames/s image sequence an average image overlap area of 88% was estimated using equation 5.10, while for the 2 frames/s image sequence the average image overlap area was 78%. With this high area overlap, one would have expected both image capture rates to give more similar results, without any large deviations, as shown in Figure 5.13.

This statement will only be true if the burst rate of the froth bubbles is very low. In general there exists an intuitive relationship between the stability and mobility of surface froth. The more mobile the froth, the lower the stability of the froth (bubbles bursting faster, etc.), while the lower the mobility of the froth the higher the stability of the froth. In general, the bursting of the froth bubbles is the highest natural dynamic response of the froth. As a result, it can be concluded that the bubble burst rate will probably be the limiting factor in choosing the optimal (minimum) image capture rate for the mobility measure.

5.5 Conclusion.

In this chapter, based on previous methods implemented, DICC was proposed as an intuitive tool to help with the characterisation of the mobility and stability of froth. By evaluating the use of image cross-correlation on both practical and video data from batch and industrial flotation cells, it was shown that image cross-correlation has the ability to measure the stability and mobility of froth. This evaluation also showed that a high image capture rate is not required to measure the mobility and stability of froth and that the capture rate must only be high enough to capture the highest natural dynamics of the froth. Thus, depending on the flotation process being monitored, the image capture rate can be minimised.

Even though DICC has a good ability to measure the stability and mobility of froth, it has the disadvantage of having high processing and memory requirements. This is further addressed in Chapter 6, where an image processing platform is discussed for measuring the stability and mobility of froth from a group of flotation cells.

CHAPTER 6

AN IMAGE PROCESSING PLATFORM FOR THE MVC.

In all MVC implementations, a suitable image-processing platform is required to interpret and extract information from the captured images. Chapter 5 indicated that mobility and stability were the two most important features of surface froth and showed that discrete image cross-correlation (DICC) is a suitable tool to extract these features. In this chapter two architectures are considered to implement the proposed DICC function of Chapter 5. The proposed architectures are based on the well-known INMOS T800 transputer and Field Programmable Gate Array (FPGA) technology. The principle function of the proposed hardware is to implement DICC through the use of the Fast Fourier Transform (FFT).

A short overview of the requirements for the image processing system is given, after which the choice of hardware explored is explained. The manner in which the DICC can be implemented with a two dimensional FFT (2D-FFT) is discussed, followed by which an explanation of how to implement a 2D-FFT through the use of a 1D-FFT is offered. The hardware architecture considered for the implementation of the DICC function is a single T800 transputer. Based on the success of this, the possibility of using a single FPGA as a 1D-FFT co-processor to improve the DICC speed is explored.

6.1 Overview.

The primary function of the image-processing platform (IPP) is to regularly extract the relevant froth features (stability and mobility) from a single flotation cell.

For industrial flotation, a plant consists of banks of flotation cells. Each of these cells is monitored by a MVC and an image-processing platform as mentioned in Chapter 1. Figure 6.1 shows a basic block diagram of the flotation cells and MVS configuration.

Every MVS (MVC and IPP) will relay information (images or feature results) to a Primary Control System (PCS). Based on the results received from the MVS's, the PCS makes the necessary control adjustments to the bank of flotation cells.

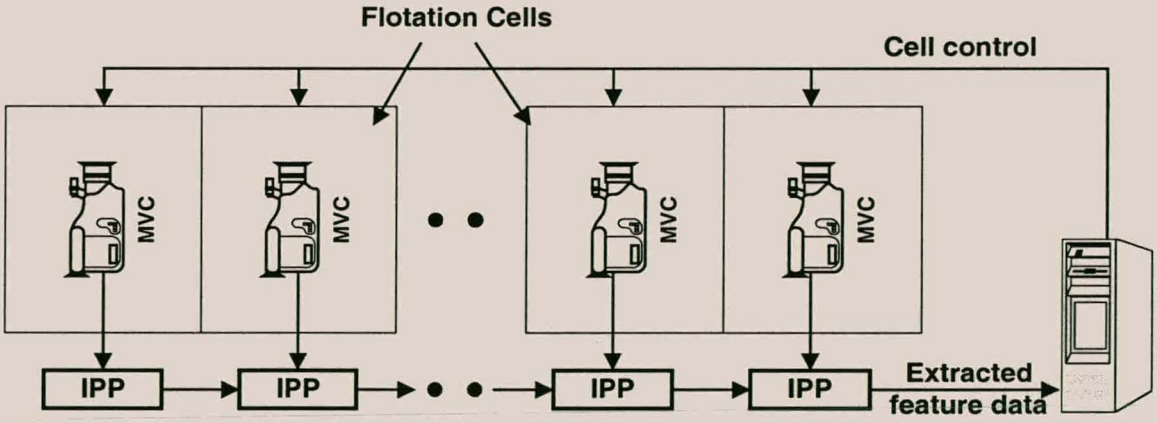

Figure 6.1 Basic block diagram of MVS configuration.

The rate at which the MVSs must supply the extracted features from the cells is dependent on the time constant of the specific flotation circuit being monitored. Paragraph 5.4.1 pointed out that the time constants for the dynamic response for a flotation plant is in the order of tens of minutes or higher depending on the specific type of flotation.

With this information in mind, a value of 1 minute was selected as the required time to implement the image cross-correlation function of Chapter 5. This value will ensure that the average mobility and stability of a flotation cell for different flotation plants can be successfully extracted. A further requirement of the MVS is that it should have the ability to relay images to the PCS for future references in order to help build a database to train plant operators.

6.2 Choice of Hardware for the IPP.

The choice of T800 transputer and FPGA technology was based on two primary considerations. Firstly, the hardware for the IPP must be as flexible as possible to allow a wide variety of image processing algorithms to be implemented on the MVSs, and not only DICC. Secondly, previous experience gained with both devices, and the availability of evaluation boards and compilers to test and verify results was taken into account.

6.2.1 IMS T800 Transputer.

The IMS T800 transputer is a 32-bit CMOS microcomputer with a 64-bit floating point unit capable of both single (32-bit) of double (64-bit) length operation, making the T800 suitable for implementing intensive mathematical functions. The IMS T800 provides a high degree of flexibility through its configurable memory interface and four standard INMOS serial communication links. The T800 can directly access a linear address space of 4Gbytes, using a 32-bit wide memory interface with multiplexed data and address lines. The on-chip configurable memory controller provides all the timing, control, and DRAM refresh signals to support a wide variety of mixed memory systems.

The standard INMOS communication links allow a network of transputers to be constructed by direct point-to-point connections, with no external logic which are capable of 5/10/20 Mbits/s serial communication. This ability allows MVSs to connect in a wide variety of network configurations which can prove very useful, depending on the type of algorithms implemented by the IPP. These communication links will also allow for the

fast transmission of flotation images to the PCS for storage. The IMS T800 transputer is available in four speed grades namely 17MHz, 20MHz, 25MHz and 30MHz. The INMOS transputer data book (SGS-THOMSON, 1989:189) provides further information regarding the IMS T800 transputer.

6.2.2 FPGA Technology.

FPGAs have opened a new path of rapid prototyping for hardware development. They provide a high degree of flexibility, while maintaining performance capabilities similar to dedicated or custom hardware. FPGAs have also proven to be suitable for implementing digital signal processing (DSP) functions; providing advantages over traditional methods (e.g. DSP processors, Application-specific standard products (ASSP) and ASIC's) through the use of parallelism, pipelining, in-circuit reconfigurability and specialised cores (e.g. 1D-FFT, discrete cosine transform etc.).

6.3 Implementing DICC with the 1D-FFT.

Equation 6.1 shows the DICC function as defined in Chapter 5.

$$p(x,y)=\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\left\{\left[IM_2(i,j)-\overline{IM_2}\right]\cdot\left[IM_1(x+i,y+j)-\overline{IM_1}\right]\right\} \qquad (6.1)$$

where M and N are the vertical and horizontal resolutions of an image respectively. IM_1 and IM_2 are two successive grey level images, while $\overline{IM_1}$ and $\overline{IM_2}$ are the average grey level values for the two images.

The DICC function of equation 6.1 is a mathematically intensive function requiring a large amount of memory and processing power. One of the most common methods of implementing DICC is through the use of the discrete Fourier transform (DFT) (equation 6.2) and the time convolution theorem as shown by equation 6.3.

$$H(n,m) = \sum_{q=0}^{M-1}\left[\sum_{p=0}^{N-1} h(p,q)e^{-j2\pi np/N}\right]e^{-j2\pi mq/M} \qquad (6.2)$$

where h is an image with size N x M pixels; and

$$r(x,y)^{\bullet} ** h(x,y) \Leftrightarrow R(n,m)H(n,m)^{1} \qquad (6.3)$$

where $**$ indicate two-dimensional convolution and $r^{\bullet}$ and h are two images. $r(x,y)^{\bullet}$ represents an image that is rotated through 180 degrees in both its rows and columns. $R(n,m)$ and $H(n,m)$ represent the 2D-DFT of the images.

In general, a straightforward computation of the discrete integral relationships of the 2D-DFT (equation 6.2) is not practical, due to the excessive number of complex multiplications and additions required. A far more efficient method is to implement the 2D-DFT with the 2D-FFT, since the 2D-FFT requires far less computations that the 2D-DFT. For a square image of N x N pixels, a computational saving factor of ½ $\log_2(N)$ for complex multiplications and $\log_2(N)$ for complex additions can be expected when using a Radix-2 based 2D-FFT. Further examination of the 2D-DFT of equation 6.2 shows that the term in brackets is simply a 1D-DFT along the data array defined by the parameter p. To evaluate the term in brackets, we compute M 1D_FFT: one for each q, along the data array defined by p. Equation 6.2 can then be rewritten as:

$$H(n,m) = \sum_{q=0}^{M-1} Z(n,q)e^{-j2\pi mq/M} \qquad (6.4)$$

[1] 2D correlation is related to 2D convolution by a 180-degree rotation of the filter matrix.

Equation 6.4 is evaluated by computing N 1D-FFT along the data array defined by parameter q. In summary, the 2D-FFT of an image is computed by first calculating 1D-FFTs along the columns of the image and then calculating 1D-FFTs along the rows of the image. In practice, since equation 6.1 represents a periodic convolution of the image $r(x, y)$ and $h(x, y)$, the image $r(x, y)$ and $h(x, y)$ in both the rows and columns must be zero padded to prevent circular convolution (Brigham, 1988:261).

In general, if a nonzero-value image of dimension (N_1, M_1) is correlated with a second nonzero-value image of dimension (N_2, M_2), the resulting function is of dimension:

$$(N, M) = (N_1 + N_2 - 1, M_1 + M_2 - 1) \tag{6.5}$$

When 1D-FFTs are involved the following additional condition must be satisfied $N = 2^{\gamma}$, $M = 2^{\eta}$ where η and γ are positive integer values. Sufficient zeros must therefore be appended to each of the images for both the rows and columns before equation 6.1 is calculated. After the use of the time convolution theorem, the resultant complex valued matrix must be transformed back to the time domain using the inverse 2D-FFT transform based on the inverse 1D-FFT. This is the same strategy followed for the forward 2D-FFT. An alternative method to implement the inverse 1D-FFT is through the forward 1D-FFT. Equation 6.6 shows this relationship.

$$h(k) = \frac{1}{N}\sum_{n=0}^{N-1} H(n)e^{j2\pi nk/N} = \frac{1}{N}\left[\sum_{n=0}^{N-1} H^*(n)e^{-j2\pi nk/N}\right]^* \tag{6.6}$$

where $*$ implies conjugation.

As a result, only forward 1D-FFT have to be implemented to do the DICC function of equation 6.1.

In summary : To compute the DICC function of equation 6.1, using only forward 1D-FFTs, the following steps are required.

1. Let IM_1 and IM_2 be two P x Q positive integer valued images.
2. Subtract from each of the two images their respective average grey level intensities.

$$I_1 = IM_1 - average(IM_1)$$

$$I_2 = IM_2 - average(IM_2)$$

3. Rotate I_1 through 180 degrees in both the rows and columns.

$$I_R = Rotate(I_1, 180°)$$

4. Augment I_R and I_2 with zeros to satisfy equation 6.5.

$$I_R(p, q) = 0 \quad p = P, P+1, \ldots, N-1 ; q = Q, Q+1, \ldots, M-1$$

$$I_2(p, q) = 0 \quad p = P, P+1, \ldots, N-1 ; q = Q, Q+1, \ldots, M-1$$

where $N = 2^{\gamma}$, $M = 2^{\eta}$ for positive integer values of η and γ.

5. Compute the 2D-FFT of I_R by first calculating N 1D-FFTs for the rows of I_R and then M 1D-FFTs for the columns of the row results.
6. Repeat step 5 for I_2.
7. Compute the array multiplication of the two images from step 5 and 6.

$$Z = I_{R(2D-FFT)} \cdot * I_{2(2D-FFT)}$$

8. Compute the inverse 2D-FFT of Z, similar to step 5 through the use of forward 1D-FFT and equation 6.6.

Additional information regarding the implementation of DICC through the use of the 1D-FFT can be found in Brigham (1988:260).

6.4 DICC on the IMS T800 Transputer.

This paragraph evaluates the ability of the IMS T800 transputer to compute the DICC function of equation 6.1. The evaluation is done on a custom T800 evaluation board similar to the standard IMS B004 evaluation board (INMOS, 1985). The evaluation board

consists of a single 20MHz IMS T800 transputer, with 2Mbytes of external DRAM and uses a IMS C002 link adaptor to interface with the ISA Bus of a PC. The compiler used is Rowley Parallel Modula-2 for the transputer (Rowley, 1992).

Only the ability of the T800 transputer to calculate 512-point and 1024-point 1D-FFTs is evaluated. Based on these results, a minimum time estimate for the 2D-FFT is made. From the 2D-FFT result, sufficient information is gained to determine the ability of the IMS T800 transputer to compute the DICC function of equation 6.1.

6.4.1 1D-FFT on the IMS T800 Transputer.

Various methods exist to calculate the 1D-FFT. For the evaluation of the T800 it was decided to implement the 1D-FFT through the use of a standard decimating in time (DIT) complex valued Radix-2 FFT (Brigham, 1988:148). Assume that the MVC as described and implemented in Chapter 3 is used. The MVC outputs 692 (H) x 494 (V) pixel colour images. Since DICC can only be implemented using monochromatic data, only one of the colour components (R, G or B) is used. This implies that the images used will only be 346 (H) x 292(V) pixels. Before the 2D-FFT can be calculated on the monochromatic image, it must be zero padded (using equation 6.5) in both its rows and columns to 1024 (H) x 512(V) pixels, as required by a Radix-2 FFT. As a result, 1024 512-point 1D-FFTs and 512 1024-point 1D-FFTs have to be calculated to implement the 2D-FFT.

Table 6.1 1D-FFT execution times (in milliseconds) on a 20MHz T800 transputer.

1D-FFT	32-bit	64-bit	% time increase
512-point	57.344	77.632	35.38
1024-point	125.376	169.664	35.32

Table 6.1 summarises the results for implementing 32-bit and 64-bit complex 512-point and 1024-point DIT Radix-2 1D-FFTs on the T800 transputer. From these results it can be seen that the execution time for a 32-bit 2D-FFT on a 1024 (H) x 512 (V) zero padded image will take approximately 123 seconds, already exceeding the 1 minute maximum

allowed for the whole DICC computation[2]. A possible solution to decrease the execution time for the FFTs could be to implement higher order Radix FFT algorithms to reduce the amount of computations required compared to the Radix-2 1D-FFT. Table 6.2 summarises the computations involved, while Table 6.3 summarises the computational savings.

Table 6.2 Number of non-trivial real multiplications and additions to compute a *N*-point complex FFT.

	Real Multiplication's				Real Additions			
N	Radix 2	Radix 4	Radix 8	Split Radix (4,2)	Radix 2	Radix 4	Radix 8	Split Radix (4,2)
512	4360		3204	3076	13566		12420	12292
1024	10248	7856		7172	30728	28336		27652

Source: Extracted from Proakis et al.1996.

Table 6.3 Computational savings of higher Radix and Split Radix complex 1D-FFT's over the Radix-2 1D-FFT.

	Real Multiplication's			Real Additions		
N	Radix 4	Radix 8	Split Radix (4,2)	Radix 4	Radix 8	Split Radix (4,2)
512		26.5 %	29.4 %		8.4 %	9.4 %
1024	23.3 %		30.0 %	7.8 %		10.0 %

From the above table it is seen that the Split Radix 1D-FFT has the highest computational savings over the Radix-2 1D-FFT. In practice, the overall savings expected when using higher Radix or Split Radix 1D-FFTs will be closer to the savings predicted for the Real additions in Table 6.3. The reason for the lower expected savings can be attributed to the similar execution times for addition and multiplications on a T800 transputer for both 32-bit and 64-bit Real values (SGS-THOMSON, 1989:210). Thus, higher Radix or Split Radix implementations of the 1D-FFT will not dramatically reduce the 123 second minimum predicted value for the 2D-FFT of a 1024 (H) x 512 (V) pixels image.

[2] For the source code of the 1D-FFT program, refer to Appendix I.

Fleury et al. (1998), found similar results when evaluating parallel implementations of the standard 2D-FFT based on the DIT 1D-FFT. Fleury et al. indicate that faster 1D-FFT can be implemented using higher order or mixed Radix FFTs. Fleury et al. conclude that the mixed Radix 1D-FFT is the fastest. Table 6.4 summarises his results for a single T805[3] transputer running at 25MHz. The last column of the table has been scaled for a 20MHz T805 transputer.

Table 6.4 Execution times (in seconds) for a mixed-Radix 64-bit complex 2D-FFT using a T805 transputer.

Image size	Radix Factorisation	Execution time	Scaled result
512 x 512	4,4,4,4,2	88.36	110.45

From this and the above results, it can be concluded that it is doubtful if the DICC function of equation 6.1 can be implemented under 1 minute for a single T800 transputer. Multiple transputer implementations have shown to be able to implement 2D-FFTs faster then a single T800 transputer, with speed-up factors approximately equal to the total number of processors used (Fleury et al., 1998), (Roebbers, 1990). The drawbacks of using multiple processors are the loss of flexibility (algorithms implemented on the IPP must now be suitable for parallel processing), the higher overall complexity of the IPP and the increase in cost of the IPP. Paragraph 6.5 looks at the possibility of using a FPGA as a 1D-FFT co-processor to help the T800 solve the DICC function less than 1 minute.

6.5 DICC using a FPGA as a 1D-FFT Co-Processor.

To alleviate the burden of the IMS T800 transputer trying to directly compute the DICC function, through the use of 1D-FFTs, a single FPGA is proposed as a 1D-FFT accelerator. The principle of the design is to allow the FPGA to calculate all the 1D-FFTs.

[3] The mathematical capabilities for the T800 and T805 are exactly the same.

6.5.1 1D-FFT with a FPGA.

Various FPGA manufacturers and third party vendors produce a variety of "cores" or ready to use solutions (Correlators, Filters, 1D-FFT transforms etc.) that can easily be implemented on their respective FPGA devices.

For this evaluation, only 1D-FFT cores for ALTERA, Xilinx and Atmel FPGA devices were considered, since they are the most well known FPGA manufacturers and are well supported by third party vendors. Table 6.5 summarises the 1D-FFTs available from these manufacturers.

Table 6.5 Summary of the 1D-FFT core features.

	ALTERA	Xilinx	Atmel
Transform size (*N*)	User defined	1024	256
FFT Method	Radix-2 (DIF)	Radix-2 (DIF)	Radix-2 (DIF)
Input Data type	Complex	Real	Complex
Data Resolution	User defined	16-bit	12-bit
Arithmetic type	Block floating-point	fixed-point	fixed-point
Memory interface	User defined	fixed	fixed
Max clock speed	41MHz (1)	2MHz	21MHz
Conversion speed	250us (2)	8.7ms	98us

1. For 16-bit data representation.
2. $N = 1024$

From Table 6.5 it can be seen that only the FFT core from ALTERA is suitable for the DICC computation. The ALTERA core is the only core capable of being configured to do 512-point or 1024-point 1D-FFTs. The core does not have a fixed data and memory interface standard, and provided a high degree of flexibility to interface with the 32-bit T800 transputer. The ALTERA core is also capable of computing complex FFTs. More information about these 1D-FFT cores can be found in the manufacturer's respective data sheets: ALTERA (1997), Xilinx (1998) and Atmel (1998).

6.5.2 The ALTERA Fast Fourier Transform.

The ALTERA MegaCore FFT function implements a decimation in frequency (DIF) algorithm and contains all the core logic functions necessary to compute a 1D-FFT. Data is input to the FFT core in normal order while the resultant data from the core is in bit-reversed order. Simple dedicated logic can be added to the core so that data always appear in normal order.

To maximise flexibility, the FFT function does not include a memory or I/O interface. The memory and I/O interface is dependent upon the end application, and thus must be customized to fit each application.

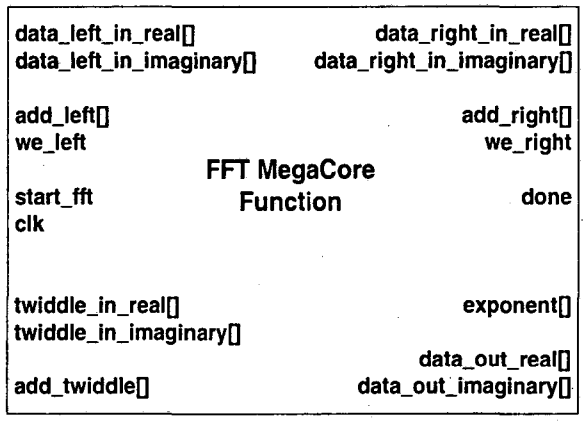

Figure 6.2 Block diagram of the FFT MegaCore Function.

Figure 6.2 shows a simple block diagram of the FFT core. To optimise throughput, the FFT function uses a dual memory architecture (Left and Right memory bank) in which data is read from one memory and written to the other. The FFT function also uses a third memory, known as the Twiddle memory, which is used to store the exponential terms required for the FFT butterfly calculations.

The number of points in the FFT determines which memory bank the FFT core reads first. When the number of points is an odd power of 2, data is always read from the right

memory on the first pass through the FFT and the final result is obtained from the Left bank. When the number of points is an even power of 2, the FFT core alternates between each memory it reads first after every complete calculation. The FFT core also enables the designer to choose the number of pipeline stages in the memory paths, allowing a trade-off between size and speed for a particular memory interface. All data buses in and out of the FFT core are in 2's complement fractional notation, with a dynamic range from –1 to 1. The FFT core also uses Block floating-point arithmetic, providing a higher dynamic range for the same bit-width fixed-point arithmetic, since scaling by ½ is only implemented in each FFT column of calculations if the results from the previous column are likely to cause overflow in the current column.

After every FFT calculation, the FFT core output the resultant Block floating-point exponent (2^{exponent}) with which the output data must be scaled. The FFT core requires $(N+15)\log_2 N$ clock cycles to perform an N-point FFT, which implies that a 1024-point FFT and 512-point FFT will take 10390 and 4743 clock cycles respectively. If the FFT core is clocked with the internal clock of the 20MHz T800 transputer it will approximately take 520μs and 237μs to calculate a 1024-point FFT and 512-point FFT respectively, exceeding the FFT capabilities of the T800 transputer by at least two orders.

Figure 6.3 shows a simplified implementation of the FFT core for an odd power of 2 FFT. Both the Left and Right memories are SRAM with a registered input, output and address busses, while the Twiddle memory is ROM with a registered output and address bus. All the registers are synchronised with the clock of FFT core. For Figure 6.3 the FFT core has been set for 3 pipeline stages for the Left and Right memory banks and 2 for the Twiddle memory. Real or complex valued data is written to the Right memory bank. After the data has been loaded, the FFT core is signalled to start the FFT calculation by pulsing the start_fft signal from Low to High to Low. As the FFT core calculates the FFT, data will be transferred an odd number of times between the two memory banks until the fft_done signal goes High to indicate completion of the FFT. The resultant complex valued data can now be read from the left memory back and be scaled by

$2^{exponent}$. Detailed information about the ALTERA FFT MegaCore can be found in the data sheet of the FFT core (ALTERA, 1997).

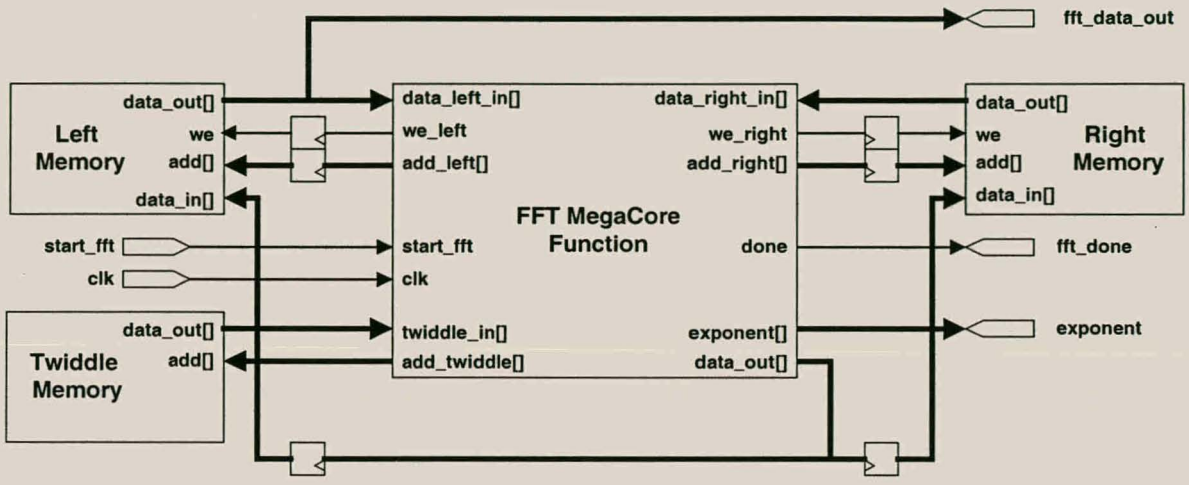

Figure 6.3 Example of implementing the FFT core for an odd power of 2 FFT.

6.5.3 Configuring the ALTERA 1D-FFT Core to Interface with the IMS T800 Transputer.

Two factors must be kept in mind when selecting a suitable bus width for the IMS T800 transputer. Firstly the interface bus must be compatible with the 32-bit bus standard of the T800 transputer. Secondly the resolution of the 1D-FFT must be high enough to minimise the numerical errors caused by quantisation.

Table 6.6 shows the expected signal to noise ratios (SNR) for the FFT core for various length FFTs with different data widths. The SNRs were determined by comparing the accuracy of MATLAB's FFT function with the results obtained from a MATLAB simulation of the FFT core, for a random uniform distributed signal with a dynamic range between –1 and 1[4].

[4] The MATLAB simulation of the FFT core is freely available from ALTERA's public ftp site (ALTERA).

Table 6.6 Expected SNR for the ALTERA FFT core.

	256-point FFT		512-point FFT		1024-point FFT	
Data width	σ_{err}	SNR (dB)	σ_{err}	SNR (dB)	σ_{err}	SNR (dB)
8-bit	4.132E-1	-2.9	8.414E-1	-9.3	1.167	-12.2
16-bit	1.501E-3	45.8	2.993E-3	39.7	4.387E-3	36.3
24-bit	5.921E-6	93.9	1.242E-5	87.3	1.742E-5	84.3
32-bit	2.347E-8	142	4.630E-8	135.9	6.933E-8	132.3

From the above table it can be seen that an 8-bit data width is not sufficient for the computation of 256-, 512- and 1024-point FFTs, while the 16-bit data width gives a SNR larger than 30dB for all three of the transform lengths. For the calculation of the DICC function of equation 6.1, the SNR for the 16-bit data width should be sufficient. The 16-bit data also corresponds to 32-bit complex numbers, which is compatible with the 32-bit data bus of the IMS T800 transputer.

Table 6.7 Available configurations in EPF10K100 and EPF10K50 (1).

Device	EABs Available	Data Width (Bits)	Twiddle Width (Bits)	Memory Implementation					
				Twiddles & Data in EABs		Twiddles in EABs & Data in External RAM		Twiddles in External RAM & Data in EABs	
				Points	EABs	Points	EABs	Points	EABs
EPF 10K100	12	≤ 8	≤ 8	512	10	2048	8	512	8
		9 to 16	≤ 8	256	10	2048	8	256	8
		9 to 16	9 to 16	256	12	1024	8	256	8
		> 16	> 16	(2)	(2)	(2)	(2)	(2)	(2)
EPF 10K50	10	≤ 8	≤ 8	512	10	2048	8	512	8
		9 to 16	≤ 8	256	10	2048	8	256	8
		9 to 16	9 to 16	(2)	(2)	1024	8	256	8
		> 16	> 16	(2)	(2)	(2)	(2)	(2)	(2)

Source: extracted from ALTERA FFT core Data sheet (ALTERA, 1997).

(1) If both the data and twiddles are in external RAM, the design can use an unlimited number of points.

(2) The device cannot support this configuration because the RAM required exceeds the available memory in the device. If these configurations are required a larger FLEX 10K device is required.

Table 6.7 shows the possible memory configurations of the FFT core for two popular FLEX 10K devices. The ALTERA FFT core is optimised for the ALTERA FLEX 10K architecture. These devices contain a finite amount of internal memory configured in Embedded Array Blocks (EABs) (ALTERA, 1996:36) which allow the core to implement some FFT computations on-chip. From the above table it can be seen that both devices support a 16-bit data width. The table also indicates that external memory will be required when computing 16-bit 512-point and 1024-point FFTs, while keeping the 16-bit Twiddle memory in embedded memory (EAB).

In practice the EPF 10K100 FPGA will be used instead of the EPF 10K50, since an 16-bit 1024-point FFT uses approximately 94% of the device resources, leaving almost no logic to implement the T800 data interface to the FPGA. The same configuration on the EPF10K100 only use 56% (3000 logical cells) of the device's resources, leaving sufficient logic to implement any additional required functions to interface the FFT core with the T800 transputer. Detailed information for the EPF10K devices can be found in the ALTERA data book (ALTERA, 1996).

6.5.4 Evaluation.

Figure 6.4 shows a simplified block diagram of the T800 transputer and FPGA configuration. The architecture for the FPGA is typical of the configuration required by the ALTERA FFT core for 16-bit FFTs. The T800 directly interfaces with the FPGA, transferring data to and from the external memory banks of the FFT core.

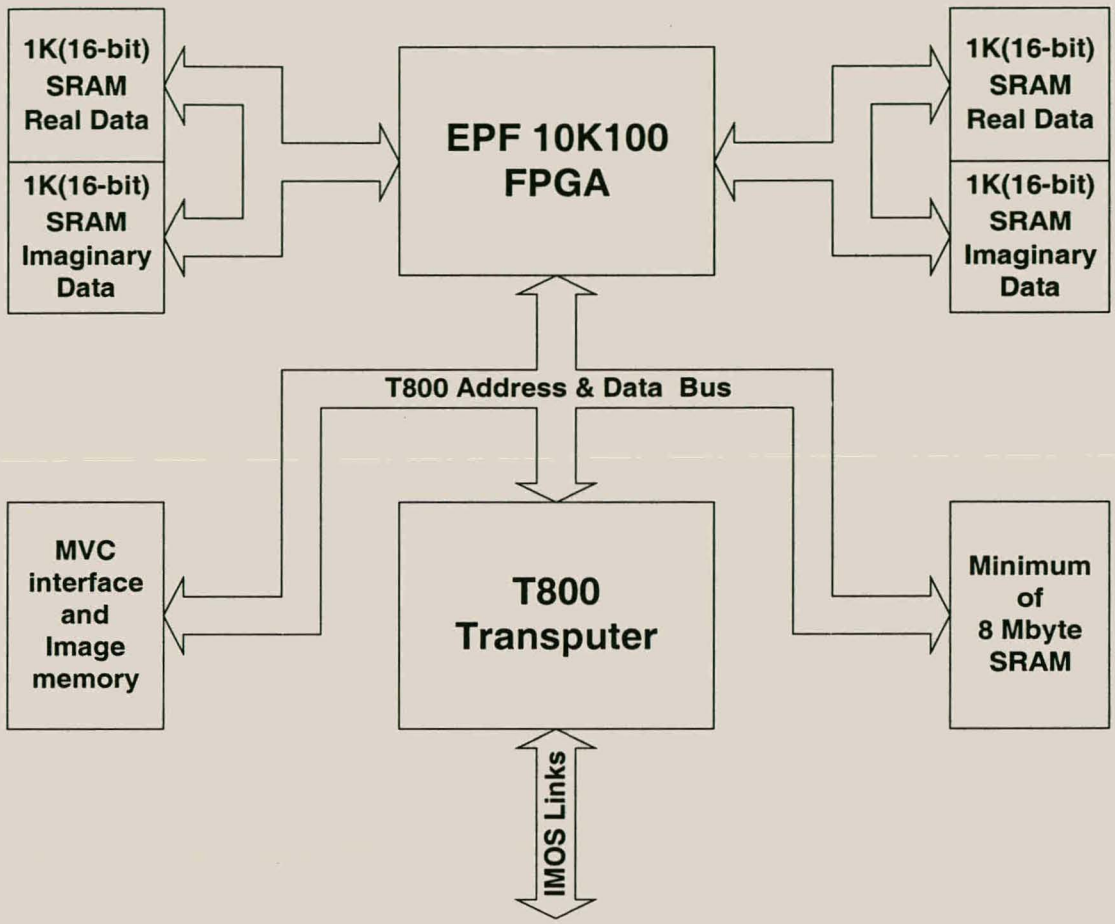

Figure 6.4 Simplified block diagram of T800 and FPGA architecture.

The operation of the architecture is as follows: The MVC interface will capture two images from the MVC and store it in external memory. The T800 transputer will then start implementing the DICC function of equation 6.1 using the steps of paragraph 6.3, except that the FPGA is computing all the required 1D-FFTs. The primary function of the T800 transputer is now to read and write complex valued data to and from the external memory of the FPGA, while continuously scaling the complex valued data between 16-bit Block floating-point notation (fixed-point notation with a variable $2^{exponent}$ scalar) and 32-bit floating-point notation. The reason for the scaling is to reduce errors caused by rounding.

From the brief discussion of the FFT core in paragraph 6.5.2, it can be seen that the core is able to calculate a 2D-FFT of a 1024 (H) x 512 (V) image in 509 ms, which is approximately 242 times faster that the T800 transputer. Unfortunately, this order of improvement will only be true if the T800 transputer is capable of transferring 16-bit complex valued data to and from the external memory of the FFT core at the same speed at which the core is capable to calculate the required FFTs. Consequently, to evaluate the expected performance of the T800 and FPGA architecture, only the time required to scale complex valued data between 32-bit floating-point notation and 16-bit Block floating-point notation is measured, while taking into account the time required to transfer 512 (16-bit) and 1024 (16-bit) complex valued data to and from the external memory of the FPGA.

The following steps describe the actions required to scale the complex valued data to and from the FPGA:

1. Scale the 32-bit floating-point complex valued data into 16-bit 2's complement fractional notation for the FPGA.

   Let's $\{C\}$ be a 512-point or 1024-point complex array, where $C(n) = x(n) + j \cdot y(n)$.
   Both $x(n)$ and $y(n)$ are 32-bit real numbers.

   r_norm = maximum[abs$\{x\}$]
   im_norm = maximum[abs$\{y\}$]
   where r_norm and im_norm are 32-bit reals

   For every element n of $\{x\}$ and $\{y\}$ do

   $$r\_temp(n) = round\{(2^{15} - 1) \cdot x(n) / r\_norm\}$$
   $$im\_temp(n) = round\{(2^{15} - 1) \cdot y(n) / im\_norm\}$$

   where r_temp and im_temp are two 32-bit integers.

Convert all negative values into 2's complement notation.

$r\_temp = r\_temp + 2^{16}$

$im\_temp = im\_temp + 2^{16}$

Output data to FPGA.

$FPGA(n) = r\_temp + 2^{16} \cdot im\_temp$

where $FPGA(n)$ is an array of 32-bit integer.

End.

2. Wait for the FPGA to calculate the FFT.

3. Scale the 16-bit 2's complement fractional notation data back to 32-bit floating-point complex valued data.

Retrieve the Block floating-point *exponent* (6-bit positive integer) from the FPGA FFT core.

For every element n of $\{FPGA\}$ do

r_temp = lower 16-bits of $FPGA(n)$

im_temp = upper 16-bits of $FPGA(n)$

Convert all 2's complement negative number back to 32-bit integers.

$r\_temp = r\_temp - 2^{16}$

$im\_temp = im\_temp - 2^{16}$

Scale data back to 32-bit reals.

$x(n) = r\_temp \cdot r\_norm \cdot 2^{exponent} / (2^{15} - 1)$

$y(n) = im\_temp \cdot im\_norm \cdot 2^{exponent} / (2^{15} - 1)$

End.

Table 6.8 summarises the times required to implement the above steps on 512-point and 1024-point complex valued data. The steps were tested on the same 20MHz T800 transputer evaluation board as used in paragraph 6.4. The T800 transputer source code for the scaling test is located in Appendix J.

Table 6.8 Time (ms) required to scale complex valued data to and from the FFT core.

512-point complex valued data	1024-point complex valued data
28.093	56.145

From Table 6.8 it can be seen that the scaling cycles for both the 512-point and 1024-point complex valued data is approximately only twice as fast as the ability of the T800 transputer to calculate a 512-point and 1024-point 1D-FFTs. This implies that a maximum speed-up factor of only 2 will be obtained over a stand alone T800 transputer, when implementing the architecture of Figure 6.2. As a result, it can be concluded that the architecture will be incapable of calculating the DICC function of equation 6.1 under 1 minute.

This poor performance can be attributed to two factors:

- The additional mathematical over-head required to switch data between 32-bit floating-point notation and 16-bit fixed-point notation.

  The time saved by not calculating 1D-FFTs on the T800 transputer has now been lost due to the additional overhead created to accommodate the two different numerical formats between the two processors.

- The low external memory access speed of the T800 transputer.

  The low external memory speed of the transputer is the result of the device's "von-noumen" architecture, since most data cycles are preceded by instruction fetches.

  The T800 transputer does have a small instruction cash, able to prefetch four instructions, but since the transputer's memory is word (32-bit) accessed, the

processor requires four instructions for every memory access, making the cash inefficient (SGS-THOMSON, 1989:195).

This same reason and the fact that real additions and multiplication have similar execution times for a transputer, is also responsible for the weak performance of the T800 transputer when computing 512-point and 1024-point complex valued 1D-FFTs.

6.6 Conclusions and Proposals.

In this Chapter, two hardware architectures based on the T800 transputer and FPGA technology, were considered to implement the DICC function of Chapter 5. The choice of the hardware was based on the high degree of flexibility obtained from both devices and the availability of suitable environment for the devices to test results with. The principle function of the proposed hardware was to implement the DICC function primarily through the use of the 1D-FFT.

The first architecture was based on the idea that a single 20MHz IMS T800 transputer was capable of implementing the DICC function under one minute. Results showed that it will take in the order of 123 seconds to implement only a third of the required 1D-FFTs, already exceeding the maximum required time.

A second architecture, which was a hybrid of the first, proposed the use of a FPGA as a 1D-FFT co-processor to the T800 transputer. For this architecture, only a maximum speed-up factor of two was expected over the first architecture, still exceeding the maximum required time of one minute for the whole DICC function.

The failure of the first architecture was attributed to the weak floating-point performance of the T800 transputer, when comparing execution times for real additions and multiplications, and the T800 transputer's low memory access rates. The failure of the second architecture was also attributed to the low memory access rates of the T800 transputer, and the additional mathematical overhead generated when trying to interface a fixed-point processor (FPGA) with a floating-point processor.

To ensure that the two architectures remain practical, it is proposed that the T800 transputer should be replaced with a fixed-point (24-bit or higher) processor with a large instruction cash or "Harvard" architecture. An alternative approach could be to replace the custom IPP with a standard PC.

CHAPTER 7

CONCLUSION, RECOMMENDATIONS AND FUTURE DEVELOPMENT.

Froth Flotation is one of the most important and valuable mineral processing technique for the recovery of low-grade and complex ores.

Recent research shows that most of the common problems found in the measurement and control of Froth Flotation plants can be solved through the implementation of MV techniques. The purpose of using MV is to integrate the measurement and control procedures through the visual characterisation of the flotation process.

In this thesis, a MVC was designed and implemented for Froth Flotation in order to assist in the further development of optimal control strategies. These strategies are based on the visual appearance of the flotation process.

7.1 Design and Implementation of the MVC.

Based on the typical type of image processing implemented in the visual characterisation of surface froth, and the visual parameters of interest (texture, froth mobility and stability etc.); a colour and monochromatic PIT-CCD sensor pair from SONY semiconductors was selected for the development of a custom MVC for Froth Flotation.

The principle design goals for the MVC are autonomous function, and the acquisition of suitable images from a wide variety of Froth Flotation processes. The methodology used for the implementation of the MVC was the use of "building blocks" supplied by SONY semiconductors to implement the sub-systems of the MVC.

The MVC as implemented requires no control actions and is capable of producing images at a rate of 20 frames/s. Images from the camera are transferred through a

standard 8-bit RS-485 interface, allowing for a simple interface between the camera and custom or existing image-processing systems. The RS-485 interface also provides for the ability of the MVC to be positioned separately from its image-processing system. Through a simple interchange of the sensor pair, the colour format of the MVC can be switched between RGB and monochromatic. This ability makes it possible to enhance the resolution of the camera at the cost of RGB colour. The camera also provides image gain and offset control. Except for the initial setup, no frequent interaction of the user is required to operate the MVC.

Evaluation of the capabilities of the MVC was also carried out. This was done through the development of a MVCI, which also functions as a post-processing tool with limited real-time capabilities, for use in the further development of froth feature extraction. Through this evaluation, it was possible to determine that the camera will be capable of capturing suitable images of surface froth. The MVCI was also used to measure the maximum EDR of the camera.

7.2 Extracting the Mobility and Stability of Froth with DICC.

Based on methods previously used to characterise the two most important froth features (mobility and stability), DICC was proposed as an additional tool for the extraction of these features. Using practical images of a batch flotation cell and footage of an industrial flotation cell, it was shown that DICC is an effective tool to quantitatively measure the mobility of froth, and can give a qualitative indication for the stability of froth. More importantly, the stability measure is able to detect fast local changes in the stability of froth.

7.3 A Custom Image-Processing Platform for the MVC.

The aim of developing a custom image-processing platform for the MVC was to implement the DICC function in less than one minute, while still allowing flexibility for the implementation of additional algorithms. Three hardware architectures were eventually considered and evaluated.

Initially, it was thought that a single 20Mhz IMS T800 transputer could achieve this aim. Results showed that this could not be done, therefore a second architecture, proposing the use of a FPGA as a 1D-FFT co-processor was considered. This also proved to be unsuccessful, and so a third architecture was considered. The failure of the first architecture was attributed to the weak floating-point performance of the transputer, and its low external memory rates. The latter also contributed to the failure of the second architecture, along with the additional mathematical overhead generated.

The third architecture, on the other hand, proved adequate for this purpose. This architecture was based on the use of a PC in conjunction with the MVCI. Results showed that through the use of a market-standard PC, the aim behind the development of the custom image-processing platform could easily be achieved. This solution also proved faster, simpler, and more cost-effective than its predecessors.

7.4 Recommendations.

To improve the image capability of the MVC to monitor fast moving froths (> 6cm/s) with minimum distortion, it is recommended that a micro-controller be added to the MVC. The function of the microcontroller is to give the MVC variable exposure control.

The 6-Watt power consumption of the MVC makes it uneconomical for MV applications that require lower power consumption. For these applications, it is recommended that the linear voltage regulators of the MVC be replaced with switch mode regulators. This will have an added advantage for the MVC, since only a single supply voltage is then required instead of the current three. The only disadvantage is that care must be taken to ensure that any additional noise introduced by the switch mode regulators will have a minimal or no effect of the imaging performance of the MVC.

7.5 Future Development.

An additional application for the MVC is as a Star sensor for the microsatellite program of the University of Stellenbosch. The SUNSAT microsatellite uses IT-CCD sensors, from SONY semiconductors, for its two Star sensors. The PIT-CCD sensors used in the MVC have improved imaging qualities over the current IT-CCD sensor used for the Star sensor. Before the MVC can be used as a Star sensor, similar modifications as recommended above must first be implemented. These modifications are required, since the SUNSAT Star sensor (Jacobs, 1995) requires exposure control and must have the absolute minimum power consumption for microsatellite applications.

The availability of new CMOS Active Pixel Sensors (APS) has opened a new field in the MV market, since these sensors provide advantages over the traditional CCD sensor used for MV applications. These advantages include (Photobit, 1998a, 1998b):

- Reduced cost and size.
  Unlike CCD sensors, CMOS APS sensors are fabricated in mainstream CMOS silicon foundries, resulting in significant cost reduction as well as allowing a high level of circuit integration (AGC, CDS, A/D, exposure control etc.) on the same chip.

- Decreased power consumption.
  CMOS APS offers a 100x reduction in system power compared to CCD sensors. CCD systems require numerous power supplies and voltage regulators for operation, CMOS APS uses a single +5V (or +3.3V), reducing power supply inefficiency.

- Random access to pixel regions of interest.
  This method of "windowing" provides an added flexibility to MV applications, which require image compression, motion detection or target tracking.

- No artifacts, smear cause by charge transfer under illumination or blooming.
  The structure and design used for CMOS APS sensors minimise and eliminate many of the typical problems experienced with CCD sensors.

- Faster frame rates.
  CMOS APS is more robust against signal pickup and crosstalk than its CCD counterparts. This allows high signal rates to the sensor. The use of "windowing" also allows faster frame rates.

- On-chip circuits to realise smart functions.
  Signal processing is integrated on-chip. In addition to the standard camera function (e.g. AGC, exposure control, etc.), higher level DSP functions can also be implemented (e.g. image stabilisation, image compression, colour coding, etc.).

Using a CMOS APS as a MVC for Froth Flotation can have significant advantages over the current implementation, since all the camera sub-systems are integrated on a single chip. As a result, it is recommended that future implementation of the MVC be based on CMOS APS technology. These sensors will provide a faster development cycle and cheaper implementation of the current MVC, while proving an improved imaging capability.

APPENDIX A

CALCULATING THE DYNAMIC RANGE OF THE ICX084AK/AL CCD SENSORS.

For any type of CCD sensor, the limiting factors of its performance are the noise of the sensor and the support electronics. The predominant sources of noise are the noise and uncertainties of the irradiance (Photon shot noise) of the surface being imaged; the noise introduced by the CCD sensor (Dark signal and Dark noise, Reset noise and Clock feed through); and the quantisation noise introduced by the analog to digital (A/D) converter.

In this appendix, the maximum expected DR for the ICX084AK/AL CCD sensors is calculated. In these calculations, the above-mentioned noise sources and noise minimisation methods are taken into account.

A.1 Dynamic Range.

The dynamic range defines the maximum signal to noise ratio of the sensor, and thus the image quality of the sensor. The dynamic range (DR) is defined by:

$$DR = 20 \bullet \log\left(\frac{Vout}{Vnoise}\right) \qquad \text{(A.1)}$$

where *Vout* is the output voltage of the CCD sensor, and *Vnoise* is the total noise voltage of the machine vision camera.

For an A/D converter, the minimum and maximum *DR* are defined by (Proakis et al., 1996:755):

$$DR_{min} = 20 \bullet \log(2^n) \qquad \text{(A.2)}$$

$$DR_{\max} = 20 \bullet \log\left(\sqrt{12} \bullet 2^{n}\right) \qquad \text{(A.3)}$$

where n is the amount bits of digitisation.

Thus, from equations A.2 and A.3, the minimum and maximum DR of the A/D converter is 48dB and 60dB for 8-bit digitisation. To maximise the total dynamic range of the MVC, the dynamic range of the CCD sensor and support electronics must be matched to the dynamic range of the 8-bit A/D converter. The ICX084AK/AL sensors (with support electronics) must therefore have a dynamic range higher than 48dB.

A.2 Photon Shot noise.

If numerous images are taken simultaneously from a uniform white surface under the same illumination conditions, there will be a slight variation in the images. This variation (noise) is caused by the random nature of the light (photons) arriving on the CCD sensor. The value of this noise is dependant on the ability of the CCD sensor to detect photons (Quantum efficiency), along with the average amount of incident photons. Simply stated, the value of the noise is dependent on the average amount of signal charge collected during the integration time. The Photon shot noise can be represented by the following equation (Buil, 1991:44):

$$\sigma_{psn} = \sqrt{N_s} \bullet CCE \qquad \text{(A.4)}$$

where

σ_{psn} = the r.m.s. voltage of the Photon shot noise;

N_s = the number of photocharges registered; and

CCE = the charge conversion efficiency of the CCD sensor (in V/electron)

To predict a value for σ_{psn}, an estimation of N_s and CCE have to be made, since these values are not quoted in the data sheets of the sensors. The following relation exists

between the number of registered photocharges and the corresponding variation in output voltage of the sensor (Buil, 1991:10):

$$\Delta V = q \bullet N_s \bullet \frac{G}{C_s} \qquad \text{(A.5)}$$

where :

ΔV = the variation in voltage between the reference level and the signal level of the CCD sensor's output.

q = the electron's charge (1.6×10^{-19} C)

G = the gain of the integrated amplifier of the CCD sensor.

C_s = the capacitance of the output diode.

The integrated amplifier generally has a gain (G) lower than unity (typically 0.8) for CCD sensors, while the capacitance of the output diode (C_s) is typically 0.1pF (Buil, 1991:11). Since both these values are unknown for the ICX084AK/AL sensors, those stated above will be used for the rest of the discussion.

Using equation A.5 and the maximum linear value of 500mV for the voltage variation (ΔV) of the CCD sensors (SONY, 1997:1085, 1102), a maximum value of 390×10^3 electrons is found for N_s. A value for CCE can be found by deducing from equation A.5 that

$$CCE = \frac{\Delta V}{N_s} = q \bullet \frac{G}{C_s} \qquad \text{(A.6)}$$

From equation A.6 a value of 1.3μV/electron for CCE is calculated.

Thus, from equation A.4, the maximum Photon shot noise will be approximately 0.8mV.

When the Dark signal and noise is known at one temperature and integration time, the Dark signal and noise can easily be calculated at another temperature (within the 75°C to -60°C range) and integration time. Using equation A.7, this relation is given by:

$$\frac{V_{Dark}(T_1, T_{IT1})}{V_{Dark}(T_2, T_{IT2})} = \frac{T_{IT1}}{T_{IT2}} \bullet e^{6374(\frac{1}{T_2} - \frac{1}{T_1})} \qquad \text{(A.8)}$$

SONY quotes a maximum peak to peak Dark noise voltage of 1mV; and a maximum Dark signal voltage of 4mV for both ICX084AK/AL CCD sensors, when tested under light obstructed conditions at 60°C. An integration time of 33ms is used (SONY, 1997:1085, 1102).

For the implementation of the CCD sensors as MVC for Froth Flotation, the sensors will generally be used at room temperature (25°C) with a typical integration time of 50ms (20 frames/s). Under these conditions, from equation A.8, it can be seen that the quoted Dark signal and noise will be lowered to 0.63mV and 0.16mV.

A.4 Reset noise.

The Reset noise is associated with the charging of the floating readout diode to a referance voltage at the output stage of the horizontal transfer register of the CCD sensor, before a new charge packet (pixel) is output. The value of this noise is given by (Buil, 1991:44):

$$\sigma_R = \frac{1}{q} \bullet \sqrt{k \cdot T \cdot C_s} \bullet CCE \qquad \text{(A.9)}$$

where

σ_R = the r.m.s. voltage of the Reset noise.

q = the electron charge (1.6×10^{-19} C)

k = Boltzmann's constant (1.38066×10^{-23})

T = the temperature in Kelvin

C_s = the capacitance of the output diode.

CCE = the charge conversion efficiency of the CCD sensor (in V/electron).

For a temperature of 25°C, using the previous values for C_s and CCE, the maximum expected Reset noise voltage will be 0.2mV. In general, Reset noise is not a problem since this form of noise is fairly constant during every period of the output signal from the CCD sensor. Implementing Correlated Double Sampling can also eliminate this form of noise (Burt, s.d.:6).

A.5 Clock feed through.

Clock feed through noise is caused by capacitative coupling of the CCD sensor's clock signals to the output signal of the sensor. This type of noise is difficult to mathematically describe, and is sensor related (DALSA, 1997b). Typically, this form of noise becomes more pronounced at high operating frequencies, and has a fixed pattern. Clock feed through can be minimized by carefully sampling the output of the CCD sensor.

A.6 Result

To determine the total noise of the CCD sensor, the quadratic sum of all the relevant noise sources are obtained (with correlated double sampling used):

$$\sigma_{total} = \sqrt{\frac{V_{Dark\_noise}^2}{2} + \sigma_{psn}^2} \qquad \text{(A.10)}$$

Thus the maximum r.m.s. noise voltage (σ_{total}) will be approximately 0.81mV. Using equation A.1, the maximum expected dynamic range of the CCD senors will be 56dB. More information on CCD sensor noise can be found in (Buil, 1991) and (DALSA, 1997b).

A.7 Conclusion.

From the discussion and evaluation of CCD sensor noise when imaging a flotation process at 25°C with a typical frame rate of 20 frames/s, it is seen that the Photon shot noise, Dark noise and Reset noise are the predominant sources of noise to be expected.

By using a noise reduction method called Correlated Double Sampling, the noise of the CCD sensor can be minimised; which in turn maximises the dynamic range of the sensor. As a result the maximum predicted Dynamic Range of the ICX084AK/AL CCD sensors would be approximately 56dB, exceeding the lower limit of 48dB set by the A/D converter.

APPENDIX B

MVC SCHEMATICS.

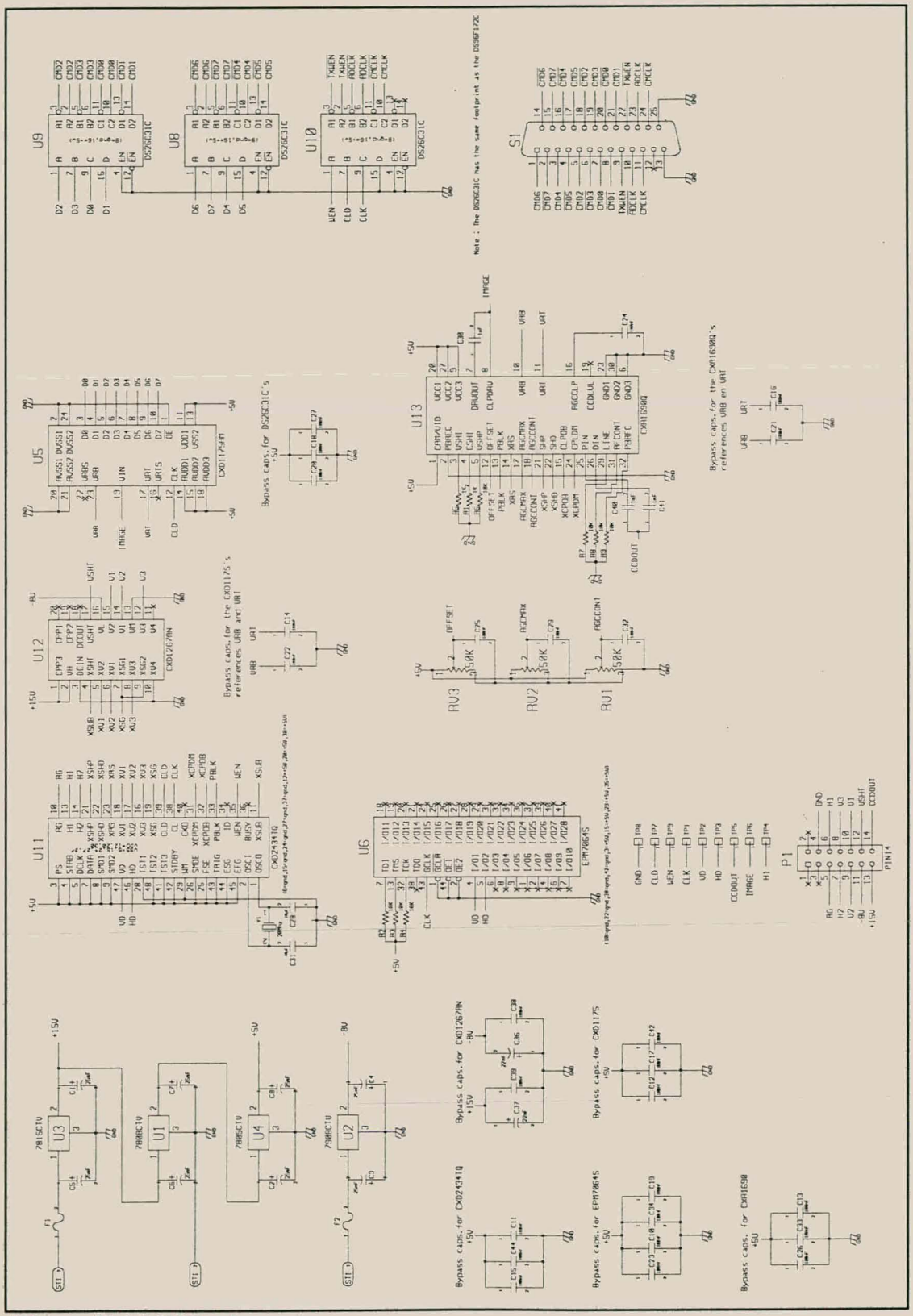

Figure B.1 MVC schematic.

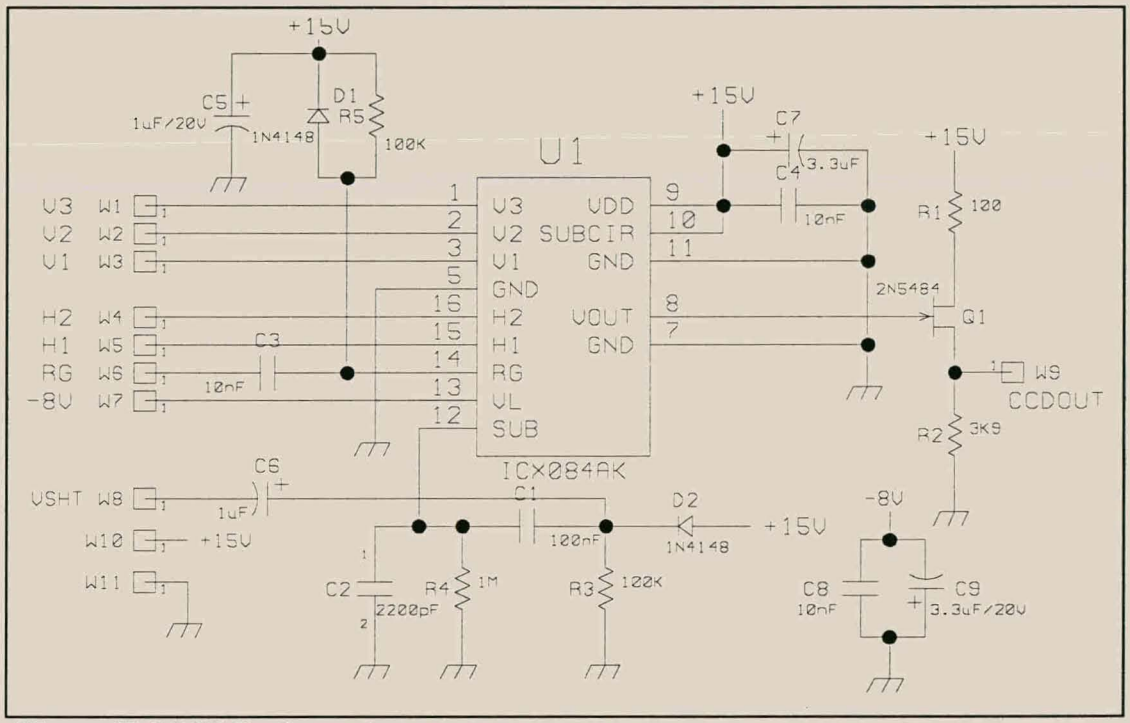

Figure B.2 MVC schematic.

APPENDIX C

VHDL SOURCE CODE FOR THE CPLD.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;

ENTITY CPLD IS
 PORT ( CLK    : IN  STD_LOGIC;
        HD, VD : OUT STD_LOGIC);
END CPLD;

ARCHITECTURE ClKgen OF CPLD IS
BEGIN
 PROCESS (CLK)
 VARIABLE CLKcount : INTEGER RANGE 1 TO 780;
 VARIABLE HDcount  : INTEGER RANGE 1 TO 525;
 BEGIN
   IF CLK'EVENT AND CLK = '1' THEN
     IF CLKcount = 780 THEN
       CLKcount := 1;
       HD <= '0';
       IF HDcount = 525 THEN
         HDcount := 1;
         VD <= '0';
       ELSE
         HDcount  := HDcount  + 1;
       END IF;
     ELSE
       CLKcount := CLKcount + 1;
     END IF;
     IF CLKcount = 78 THEN
       HD <= '1';
     END IF;
     IF HDcount = 10 THEN
       VD <= '1';
     END IF;
   END IF;
 END PROCESS;
END ClKgen;
```

APPENDIX D

IMPLEMENTATION OF A SIMPLE INTERFACE FOR THE MVC.

In this Appendix, a simple Interface is developed and implemented for the MVC. The main tasks of the Interface are to capture images from the MVC for evaluation purposes and to function as a post-processing tool with limited real-time capabilities for the further development of froth feature extraction algorithms.

For the implementation of the Interface, it was decided to use a PC as the image-processing platform for the MVC. The Interface will simply be a card that slots into the ISA (Industry Standard Architecture) Bus of the PC. The card will be able to capture images from the MVC at rate of 20 frames/s (approximately 7.7Mbytes/s), and transfer the images through the ISA Bus to the memory of the PC for image processing.

D.1 MVC Interface Implementation.

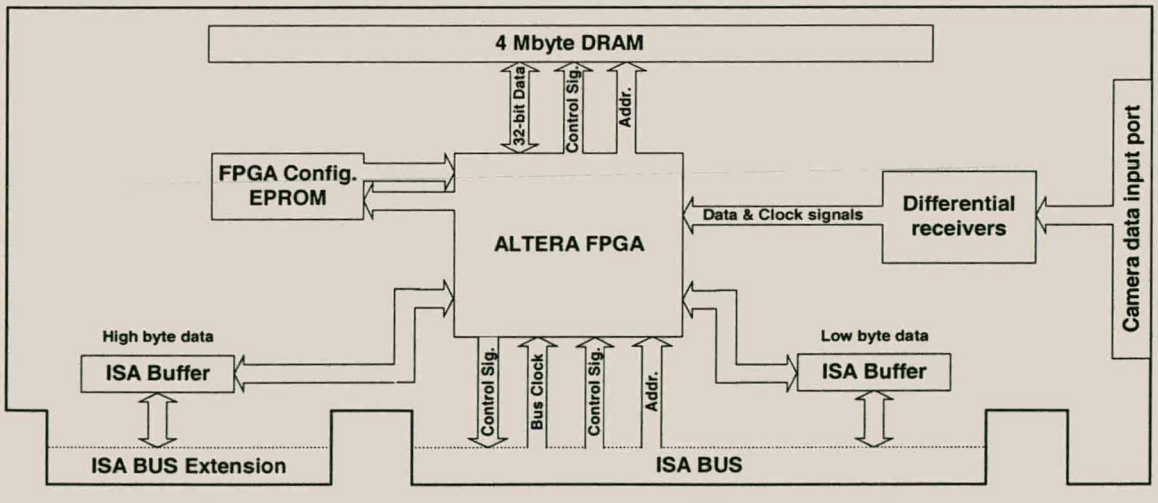

Figure D.1 Block diagram of the MVC Interface card for a PC.

Figure D.1 shows the implementation of the Interface card. Chapter 3 illustrated that the MVC requires no control signals to function, and will run autonomously from the Interface card. Differential image data and clock signals are continuously transmitted to the MVC Interface (MVCI) by the MVC and converted back to TTL levels through the differential receivers. The Field Programmable Gate Array (FPGA) writes the images from the MVC to dynamic memory (DRAM) for storage or reads images from the DRAM and downloads them through the ISA Bus. The Buffers are used to drive data between the FPGA and ISA Bus.

D.2 ISA (Industry Standard Architecture) Bus.

There are two bus standards available which are commonly found on the same PC motherboard; namely the ISA, and PCI (Peripheral Component Interconnect) Bus. Today, the PCI Bus is the industry standard capable of much higher data bandwidths in comparison with the ISA Bus. The PCI Bus has peak data transfer rates of 264Mbytes/s (32-bit) or 528Mbytes/s (64-bit) at 66MHz, which are a minimum of 40 times faster than the ISA Bus. When the overall cost, complexity and development time of implementing custom interface boards for the ISA and PCI Bus are compared, the PCI Bus is by far the more time consuming, complex and costly to use.

Previous experience with the ISA Bus showed that I/O mapped 16-bit polled data transfer is an efficient way of transferring large amounts of data over the ISA Bus. This method is capable of sustaining a data rate of 2.4Mbytes/s on a Pentium 166MHz processor, with the ISA Bus speed set at a maximum of 11Mhz. Higher rates are possible if a faster processor is used, since the data rate is primarily dependent on the processor speed for a fixed ISA Bus speed.

The size of an image received from the MVC is 385.32Kbytes large. This implies that with an ISA Bus data rate of 2.4Mbytes/s, an image download rate of approximately 6.5 images/s can be achieved. Consequently, it will take approximately 150ms to download an image using the ISA Bus.

D.3 Dynamic Memory (DRAM).

As previously discussed, the functions of the MVCI card are to evaluate the MVC and make it possible to use the MVC to monitor a flotation cell for post-processing purposes. Therefore, the MVCI has to be able to capture 385.32Kbytes images at a maximum image rate of 20 frames/s, which place a constraint on the speed and size of the storage memory.

If the price, speed, complexity, and densities of Static memory (SRAM) are compared to DRAM; it is found that DRAM has much higher densities at lower prices for the same speed SRAM, while the controlling complexity of DRAM remains higher. To make the MVCI card versatile with respect to its memory, the card was designed to accommodate a 1Mbyte (2 full images) or 4Mbyte (10 full images) DRAM module (SIMM). For the implementation of the MVCI card a 4Mbyte DRAM SIMM was used.

D.4 Field Programmable Gate Array (FPGA).

Through the use of a Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL), hardware can easily be synthesised and implemented on high-density gate devices. Various high-density gate devices are available, of which FPGAs are ideal for applications that require a large number of registers and I/O pins. The low cost and reconfigureability of SRAM-based FPGAs make them efficient vehicles to rapidly develop and implement hardware. These devices are commonly used in custom applications as PC add-on cards, multi-purpose telecommunication cards etc. (ALTERA, 1996:9).

For the development and implementation of the MVCI card, a high I/O device was needed, due to the high I/O requirements of the DRAM and ISA Bus. A 160-pin surface mount EPF8636 FPGA from ALTERA was used. The device has 114 user I/O pins, 4

dedicated input pins, 636 flip-flops and 6000 usable gates. More information on the FPGA can be found in the relevant data sheets of the device (ALTERA, 1996:150).

D.5 VHDL Implementation of the MVCI Card.

Figure D.2 shows a functional block diagram for the VHDL implementation of the FPGA on the MVCI card.

The interface card supports three configuration modes:

- *Default* mode

  The *default* mode is used to configure the DRAM address, for capturing and downloading of images, and to set the amount of images to be captured.
- *MVC to DRAM* mode

  The *MVC to DRAM* mode is used to sequentially capture images from the MVC and store them in DRAM.
- *DRAM to ISA Bus* mode

  The *DRAM to ISA Bus* mode is used to download images from the DRAM through the ISA Bus.

In the *MVC to DRAM* mode, the MVC State machine captures images (8-bit data) from the MVC and packs the data into the 32-bit DRAM Buffer. When the third byte of data is written to the buffer, the DRAM State machine is started. This will ensure that when the DRAM Buffer is filled with the fourth byte, the whole buffer is written to the DRAM before the next valid data byte is received from the MVC. This process will continue until all the images from the MVC have been captured.

In the *DRAM to ISA Bus* mode the ISA State machine controls the reading of image data from the DRAM. The ISA State machine controls the ISA DRAM Buffer, and always ensures that the buffer is full of valid image data. Data is read from the buffer 16-bits at a time. When the last 16-bits are read, the DRAM State machine refills the buffer. Since

the read cycle of the ISA Bus is slower than the execution cycle of the DRAM State machine, the State machine is always able to refill the ISA DRAM Buffer before the next ISA read cycle begins.

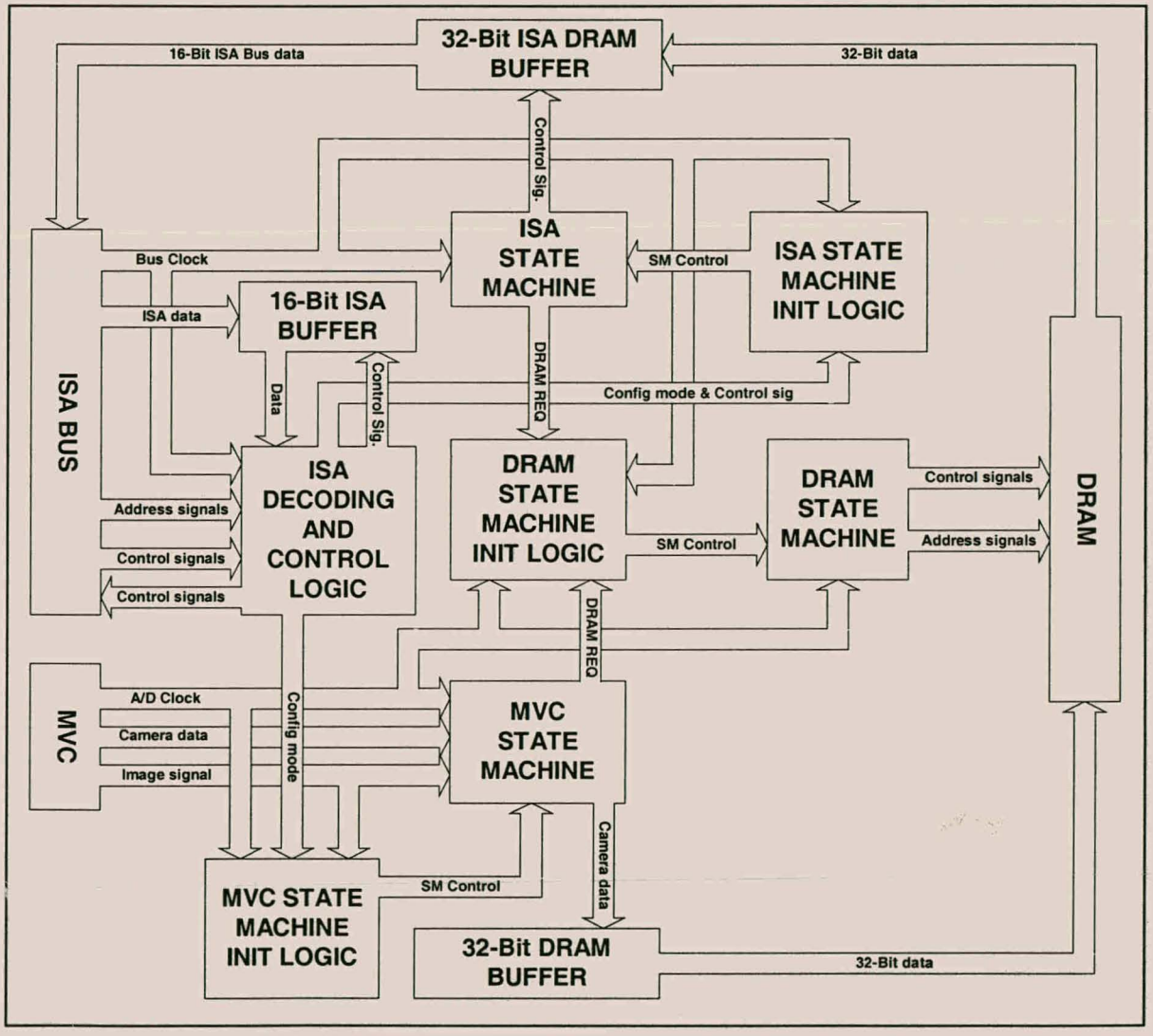

Figure D.2 Block diagram of the VHDL implementation of the MVCI card.

D.5.1 The ISA Decoding and Control Logic Block.

The *ISA decoding and control logic block* is responsible for the following functions:

- The mapping of the MVCI card into the I/O space of the PC.
- Setting the configuration modes of the MVCI card.
- Controlling the 16-bit data access to and from the ISA Bus.
- Setting the amount of images to be captured.
- Indicating that all the requested images were captured.

The MVCI is mapped into the I/O address space of the PC from address 300H to 306H. Table D.1 summarises the function of each I/O address.

Table D.1 Summary of I/O addresses and their function.

I/O	Function
300H	Configuration port: Used to set the configuration mode of the MVCI card.
301H	Image counter port: Used to set the amount of images to capture by the MVCI card.
302H	DRAM access port: Used to read images from the DRAM of the MVCI card.
303H	Polling port: Used to determine if all the requested images were captured by the MVCI card.
304H	DRAM address port: Used to set the lower 16-bit start address for the DRAM of the MVCI card.
306H	DRAM address port: Used to set the higher 6-bit start address for the DRAM of the MVCI card.

D.5.2 The DRAM State Machine.

The function of the DRAM State machine is to control the

- Writing and reading cycles of, 32-bit, data to and from the DRAM.
- DRAM address lines.
- Refreshing of the DRAM.

The data clock of the MVC determines the highest clock rate at which data will be written to the DRAM. To simplify and minimise the DRAM access cycles, only 32-bit data will be read or written to the DRAM using the standard read and write cycle of the

DRAM (HITACHI, 1994:12-13). This implies that the maximum DRAM access rate will be four times lower that the maximum MVC data rate. Consequently, a maximum DRAM access rate of 2.05MHz (8.19MHz/4) will be achieved. To further simplify the design a refresh cycle was added after every DRAM access cycle.

To ensure adequate synchronisation with the image data from the DRAM Buffer, the DRAM State machine is clocked by the A/D clock received from the MVC[1]. This implies that the DRAM access cycle plus refresh cycle must be lower or equal to four A/D clock cycles in order to continuously write image data to DRAM.

For the implementation of the DRAM State machine, care had to be taken to ensure that no glitches can occur on the row (RAS) and column (CAS) control lines of the DRAM. This is due to the fact that the DRAM is edge-sensitive with respect to these lines.

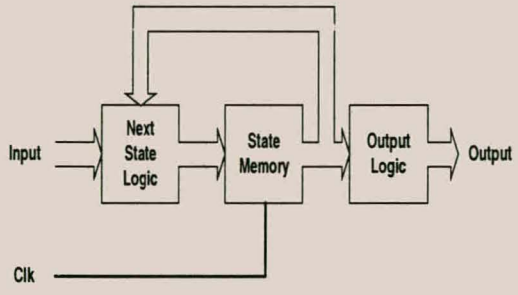

Figure D.3 Conventional State machine structure.

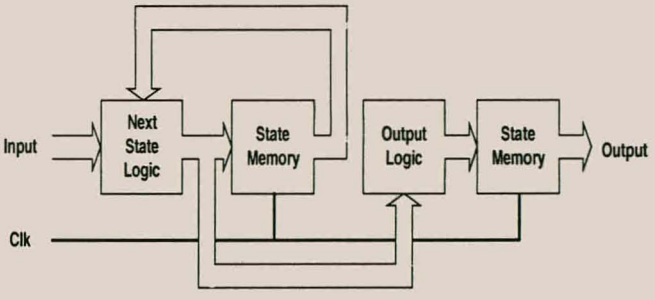

Figure D.4 Adapted State machine structure.

Figure D.3 shows the typical structure of a State machine. A problem with this type of State machine is that the output comes from combination logic. In general, depending on the application, this is not always a problem, but when a State machine is implemented in a FPGA, the user has no real control of the way the output of the state memory is routed to the combination logic. This implies that the propagation delays from the state memory to the combination logic are not guaranteed to be the same, and any state changes will

[1] The A/D clock received from the MVC is the CL clock from the Timing Generator of the MVC.

cause glitches on the output. To overcome this problem, a more suitable State machine structure as shown in Figure D.4 was used.

The differences between the two structures are that the adapted State machine has two state memories and its output comes directly from the secondary state memory. The conventional State machine has only one state memory and its output is based on combination logic of the state memory. The advantage of the adapted State machine is that the output of the State machine has no glitches. The disadvantage is the use of additional state memory.

To ensure that the DRAM State machine is able to complete in four A/D or fewer clock cycles, the State machine is triggered by both the positive and negative edge of the A/D clock.

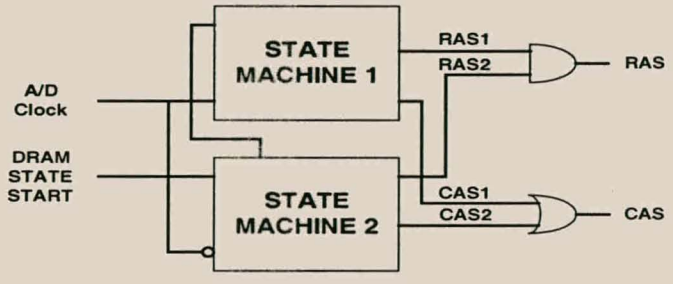

Figure D.5 The DRAM State machine structure.

Figure D.5 shows the DRAM State machine structure. Two State machines with the same structure as Figure D.4 were implemented. The first of these is sensitive to the positive edge, and the other to the negative edge of the of the A/D clock. As shown in Figure D.5, the individual row (RAS1&2) and column (CAS1&2) control lines are respectively AND'ed and OR'ed to create the actual row (RAS) and column (CAS) control lines. The combination logic on the output can never cause glitches since the output of each State machine changes with the opposite edge of the A/D clock.

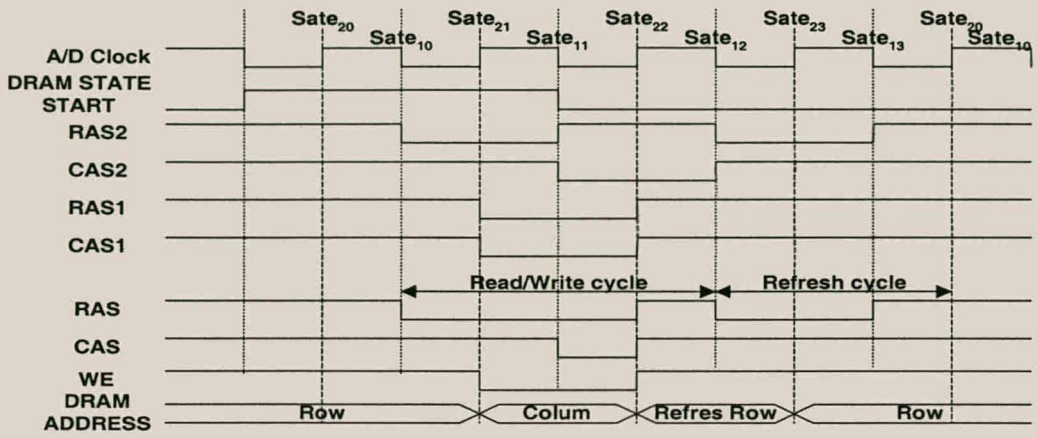

Figure D.6 A functional timing diagram for the DRAM State machine.

Figure D.6 shows the functional timing diagram of the DRAM state machine. All the data of the DRAM must be refreshed every 32ms. For a 4Mbyte DRAM this implies that every 15.6µs (64KHz) a refresh cycle has to occur (HITACHI, 1994:1). In general, when data is read from or written to DRAM, the State machine executes faster than 64KHz. This ensures that the DRAM is sufficiently refreshed.

Problems can occur when there are long periods between DRAM access cycles. To overcome this, the DRAM State machine can perform a refresh cycle similar to the one shown in Figure D.6. This occurs on the condition that sufficient time has passed since the last refresh or DRAM access cycle.

D.5.3 MVC State Machine.

The function of the MVC State machine is to continuously capture 8-bit image data from the MVC and pack it into a 32-bit DRAM Buffer. Once the buffer is filled, the State machine starts the DRAM State machine, which latches the data from the buffer into DRAM.

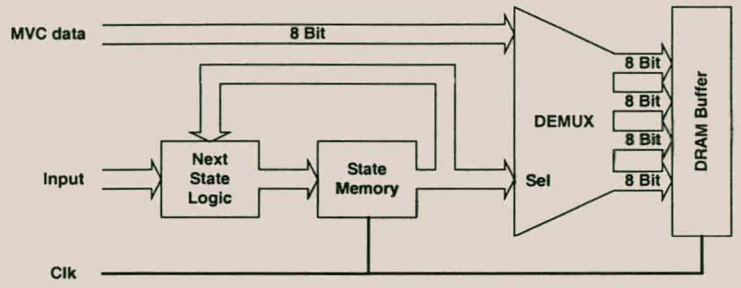

Figure D.7 The MVC State machine structure.

From Figure D.7, it can be seen that a similar State machine structure as shown in Figure D.4 is used. This structure will ensure that image data from the MVC is correctly written to the DRAM Buffer, unaffected by glitches caused by state changes of the state memory and the propagation delay of the Demultiplexer (DEMUX).

D.5.3.1 Detecting and Image.

Since the MVC runs autonomously, detection of exactly when valid image data is available to start the MVC State machine is required. This will prevent the MVCI card from capturing incomplete or invalid image data.

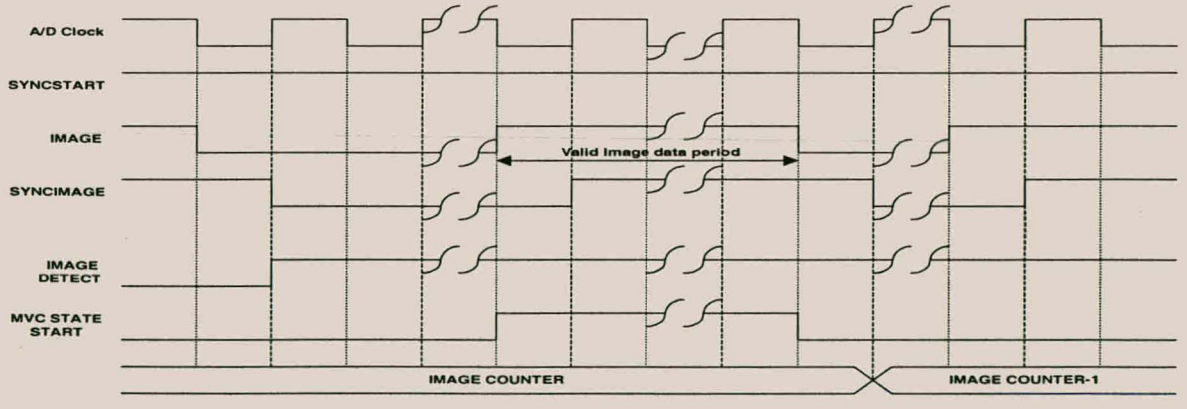

Figure D.8 A functional timing diagram for starting the MVC State machine.

Figure D.8 shows a functional timing diagram for detecting valid image data and initialisation of the MVC State machine. To ensure good synchronisation with the image

data, the MVC State machine is clocked by the falling edge of the A/D clock received from the MVC (Chapter 3).

Before the MVC State machine is able to start capturing image data, the MVCI card has to be configured to the *MVC to DRAM* (SYNCSTART being high) mode. The IMAGE COUNTER must also be set to a value of one or higher, and a previous image had to be detected (IMAGE DETECT going high). The High periods of the IMAGE signal[2] indicate when valid image data is being received from the MVC. To detect whether a complete image has been output by the MVC, the IMAGE and SYNCIMAGE signals are checked on the rising edge of the A/D clock[3]. If IMAGE is Low and SYNCIMAGE is High, the end of a valid image was detected. Image data will then be continuously captured on the falling edge of the A/D clock while the IMAGE signal is High. During this period, the MVC State machine is continuously activated (MVC STATE START signal being High). After a valid image was captured, the IMAGE COUNTER is decreased.

D.5.3.2 Starting the DRAM State Machine.

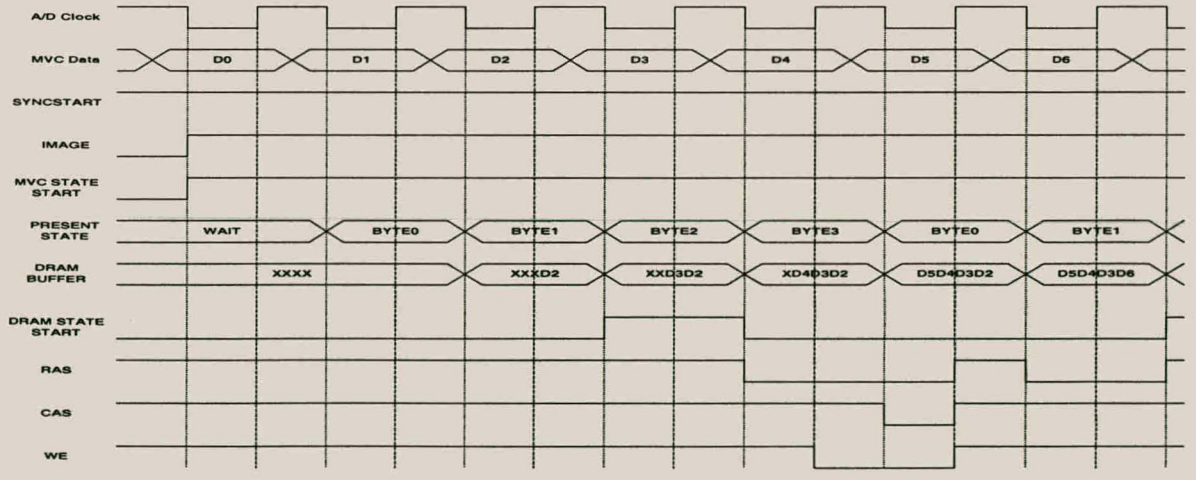

Figure D.9 The full functional timing diagram for the MVC State machine.

[2] The IMAGE signal is actually the WEN signal from the Timing generator (Chapter 3).
[3] SYNCIMAGE is half an A/D clock period delayed version of the IMAGE signal.

Figure D.9 shows the full functional timing diagram for the MVC State machine and its interaction with the DRAM State machine. As previously mentioned, once the MVCI card is configured to capture images and a previous image was detected, valid image data can be captured on the falling edge of the A/D clock while IMAGE is High. To ensure that no image data is lost, data is placed consecutively and uninterrupted byte for byte into the DRAM Buffer.

This poses a problem, since when the DRAM Buffer is full, the buffer has to be written to DRAM before the next data byte is written to the buffer. This problem can however be overcome by starting the DRAM State machine at the correct time. This ensures that when the buffer is filled, the buffer is written to DRAM without affecting its filling process. For these reasons, the importance of synchronising the MVC and DRAM State machine with the same clock (A/D clock) becomes clear.

From Figure D.6 it can be seen that the DRAM State machine requires two A/D clock cycles from initiation before data is written to DRAM. Thus, the MVC State machine must initialise the DRAM State machine two A/D clock cycles before the DRAM Buffer is filled. Figure D.9 shows that by initialising the DRAM State machine on the state change from BYTE 1 to BYTE 2 of the MVC State machine, the DRAM State machine will latch the DRAM Buffer data on the falling edge of CAS (HITACHI, 1994:13). In practice, this timing is not too critical, since the time difference between the DRAM Buffer being output to the DRAM and CAS going low will be a minimum of 10ns. This value exceeds the 0ns minimum set-up time required by the DRAM SIMM.

Figure D.9 also shows that the first two bytes of the image data stream are not captured. This occurs due to the fact that the MVC State machine requires two A/D clock cycles before the first image byte is captured. Fortunately, the first image bytes received from the MVC are dummy bits which are not valid image data (paragraph 2.10). To determine whether all the requested images were captured, the Polling port of the MVCI card is polled. If a value of one is returned, the captured images can be downloaded through the

ISA Bus. Appendix E contains an example of Turbo Pascal 7 code for the capturing and downloading of images from the MVCI card.

D.5.4 ISA State Machine.

The function of the ISA State machine is to initialise the DRAM State machine to ensure that the 32-bit ISA DRAM Buffer is always filled, and to transfer 16-bits of the buffer to the ISA Bus. Figure D.10 shows the basic structure of the State machine.

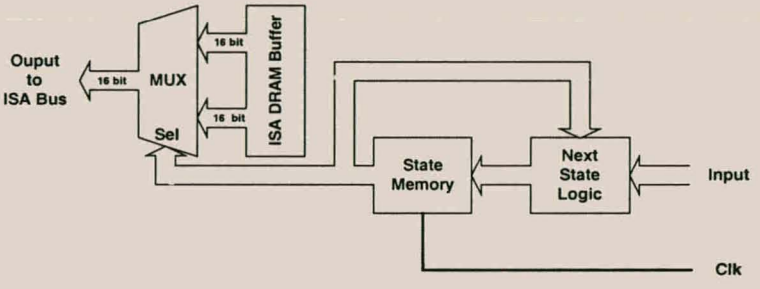

Figure D.10 The ISA State machine structure.

Figure D.10 illustrates that a similar State machine structure to Figure D.3 is used. The State machine does not have to be glitchless, since the maximum frequency of the ISA Bus read cycle is lower than the cycle speed of the ISA and DRAM State machine combination. As a result, the output of the multiplexer (MUX) will always be stable before the ISA Bus reads it. To simplify and ensure good synchronisation with the ISA Bus, the ISA State machine is synchronised with the rising edge of the ISA Bus clock (Blk).

D.5.4.1 Down Loading Images through the ISA BUS.

To download image data from the MVCI card, the Polling port has to be checked to ascertain whether the requested amount of images is available. Following this, the card must be configured to the *DRAM to ISA* mode. When this mode is set, the ISA State machine automatically starts, and requests the DRAM State machine to fill the ISA

DRAM Buffer. Since the DRAM State machine cycle is faster than the read cycle of the ISA Bus, the ISA DRAM Buffer will be filled before an ISA Bus read cycle is performed.

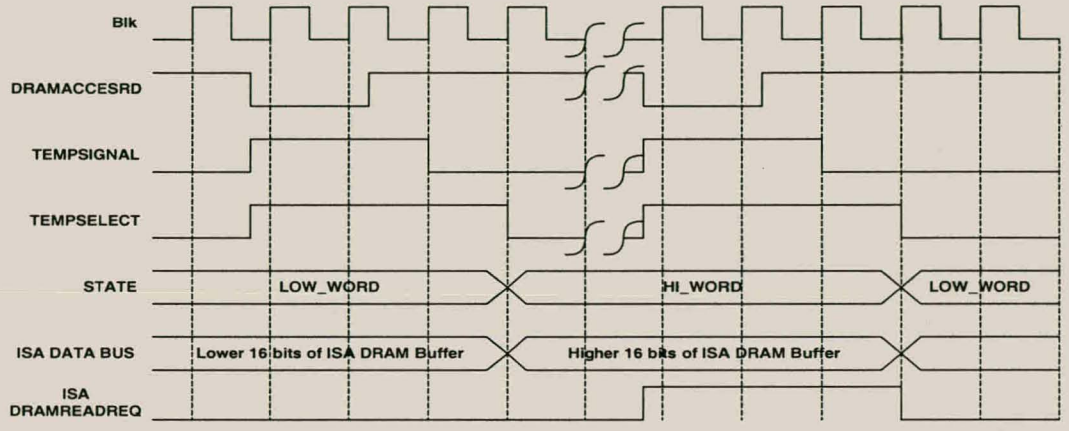

Figure D.11 A functional timing diagram of the ISA State machine.

Figure D.11 shows the functional timing of the ISA State machine. When a read cycle is performed (DRAMACCESRD is Low), the ISA State machine first places the lower 16-bits of the ISA DRAM Buffer on the data lines of the ISA Bus. The ISA Bus on the rising edge of DRAMACCESRD signal will latch the data. The state of the ISA State machine is only changed after a read cycle is performed. This is indicated by the DRAMACCESRD and TEMPSELECT signal being High, and the TEMPSIGNAL signal being Low on the rising edge of Blk.

When the upper 16-bits of the ISA DRAM Buffer are about to be read, the ISA State machine requests that the DRAM State machine refill the ISA DRAM Buffer before the next read cycle is performed (ISA DRAMREADREQ being High). As a result, the ISA DRAM Buffer is refilled after every two ISA Bus read cycles.

D.5.4.2 Synchronising the ISA and DRAM State Machine.

The problem arises that the ISA State machine is synchronised with the ISA Bus clock (Blk) while the DRAM State machine is synchronised with the A/D clock received from the MVC. Since the only assumption can be made about the relation between the Blk and A/D clock is that the A/D clock will be approximately 8MHz and the Blk will be in the range of 8-11MHz, a method had to be found to synchronise the DRAM start request signal (ISA DRAMREADREQ from the ISA State machine) with the A/D clock of the DRAM State machine. This is required in order to avoid any metastable behavior from the DRAM State machine.

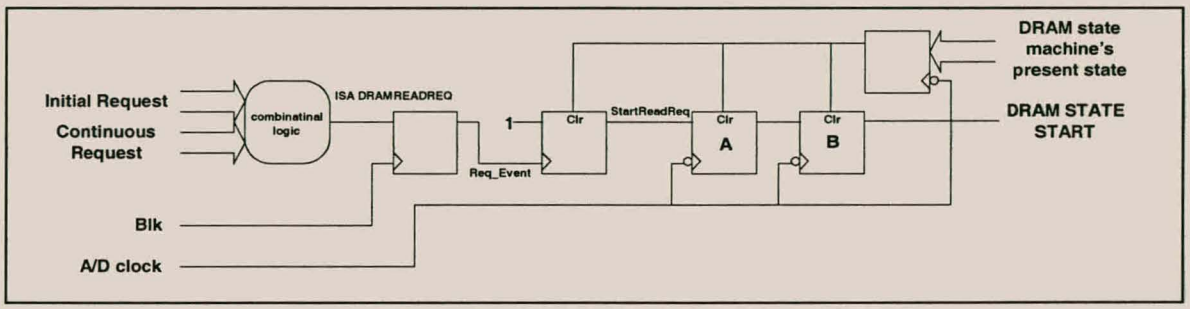

Figure D.12 Block diagram for creating an A/D clock synchronised version of the ISA Bus read request to start the DRAM State machine.

Figure D.12 shows the basic structure of the synchroniser, when the MVCI card is configured in the *DRAM to ISA* mode. As mentioned above, there are two conditions that will cause the ISA State machine to request a DRAM access cycle. The first is when the configuration mode of the MVCI card is changed to *DRAM to ISA* (Initial Request). The second when the ISA DRAM Buffer has to be refilled (Continuous Request).

When either of the two requests becomes true, the ISA DRAMREADREQ signal is cycled as shown in Figure D.11. This request is then latched on the positive edge of Blk and used to clock a new request (StartReadReq) through to a Dual-Rank synchroniser (represented by the two flip-flops labeled A and B) (Becke, 1997). The synchroniser is clocked with the negative edge of the A/D clock. The Dual-Rank synchroniser will

generate an A/D clock synchronised DRAM start signal (DRAM STATE START). Once the DRAM State machine has started the DRAM STATE START signal is reset and the DRAM Buffer is refilled on the rising edge of the A/D clock while CAS is Low. The VHDL source code of the MVCI in Appendix F, and the schematics of the MVCI in Appendix G contain more detailed information regarding the implementation of the MVCI card.

D.6 Conclusion.

Using VHDL and FPGA technology, a simple PC MVCI card was developed and implemented for the MVC. The MVCI card can capture a maximum of 10 full images continuously from the MVC at a rate of 20 frames/s (7.7Mbytes/s).

As a result, the MVCI card is a simple tool to evaluate the MVC, and can function as a post-processing tool with limited real-time capabilities for the further develop of froth feature extraction algorithms.

APPENDIX E

TURBO PASCAL 7 SOURCE CODE EXAMPLE FOR INTERFACING WITH THE MVCI CARD.

```
PROGRAM MVCI;
{$R+}
USES CRT;

VAR
  Loop       : LONGINT;
  Num        : WORD;
  LUS        : WORD;
  COUNT      : WORD;
  DF         : FILE OF WORD;
  NAME       : STRING;

BEGIN
  Clrscr;
  WriteLn('....Setting CONIGMODE to  default....');
  PORTW[$300] := $0000;

  WriteLn('....Setting DRAMLADDR....');
  PORTW[$306] := 0;

  WriteLn('....Setting DRAMHADDR....');
  PORTW[$304] := 0;

  WriteLn('....Setting IMAGECNT => 1 ....');
  PORT[$301] := 1;

  WriteLn('....Setting CONIGMODE to capture camera data....');
  PORTW[$300] := $0003;

  REPEAT
    Writeln(PORT[$303]); { Pole to see if all the requested images }
  UNTIL PORT[$303] = 1;  { where captured to DRAM. }
```

```
    Clrscr;
    Write('Enter file name => ');
    READLN (NAME);
    ASSIGN(DF,NAME);
    REWRITE(DF);
    Writeln('Creating Image File....');

    WriteLn('....Setting CONIGMODE to  default....');
    PORTW[$300] := $0000;

    WriteLn('....Setting DRAMLADDR....');
    PORTW[$306] := 0;

    WriteLn('....Setting DRAMHADDR....');
    PORTW[$304] := 0;

    WriteLn('..Setting CONIGMODE to read camera data fromDRAM..');
    PORTW[$300] := $0001;

    FOR LOOP := 1 TO 192660 DO  {780*494/2}
    BEGIN
      Num := PORTW[$302]; { Read image data from MVCI }
      Write(DF,Num);      { Save image data to file   }
    END;
    Close(DF);
END.
```

APPENDIX F

VHDL SOURCE CODE FOR THE MVCI CARD.

```
--*************************************************************************
--* NAME    : Colour CCD Camera Controller and Interface                  *
--* NOTES   : This program implements 16 bit I/O for the ISA bus with     *
--*           minimum wait states.                                        *
--*************************************************************************
--* USE     : Address 300H -> Configuration Select                        *
--*                           1) 0000H : Default                          *
--*                           1) 0001H : DRAM   -> ISA Bus                *
--*                           2) 0002H : DRAM   <- ISA Bus : Not Used     *
--*                           3) 0003H : Camera -> DRAM                   *
--*           Address 301H -> Set the Image Counter                       *
--*                           The Image Counter indicates how many        *
--*                           Images will be stored in DRAM               *
--*           Address 302H -> DRAM Access                                 *
--*           Address 303H -> Polling port                                *
--*                           When Bit0 of the port value is set to 1     *
--*                           all Images are clocked into DRAM            *
--*           Address 304H -> Set DRAM Address Counter H value            *
--*           Address 306H -> Set DRAM Address Counter L value            *
--*                                                                       *
--* NOTE    : The I/O port only supports 16-Bit data access when even     *
--*           addresses are used. 8-Bit otherwise.                        *
--*************************************************************************

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY CAMCON IS
    PORT ( -- Input Signal from the ISA Bus --
           AEN, N_IOWC, N_IORC : IN STD_LOGIC;
           SA                  : IN STD_LOGIC_VECTOR(9 DOWNTO 0);
           PCDAT               : INOUT STD_LOGIC_VECTOR(15 DOWNTO 0);
           BCLK                : IN STD_LOGIC;
           -- CAMERA Signals--
           ADCLK               : IN STD_LOGIC;
           IMAGE                       : IN STD_LOGIC;
           CAMDAT                      : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
           -- DRAM DATA --
           DRAM                : INOUT STD_LOGIC_VECTOR(31 DOWNTO 0);
           -- ISA Bus control lines --
           NOWS, N_IO16        : OUT STD_LOGIC;
           -- DRAM Address --
```

```
            DRADDR              : OUT STD_LOGIC_VECTOR(10 DOWNTO 0);
            -- DRAM's row and colum select lines --
            N_RAS, N_CAS        : OUT STD_LOGIC;
            -- DRAM's Write Strobe --
            N_DRWR              : OUT STD_LOGIC;
            -- Data buffers to ISA Bus control signals --
            N_BEN, DIR          : OUT STD_LOGIC);

END CAMCON;

ARCHITECTURE CAMISA OF CAMCON IS
TYPE   DRAM_STATE_TYPE    IS (IDLE,RAS,CAS,RF,RFCAS);
TYPE   CONTROL_STATE_TYPE IS (WAITSTATE,BYTE0,BYTE1,BYTE2,BYTE3);
TYPE   ISA_STATE_TYPE     IS (LOW_WORD,HI_WORD);
SIGNAL STATE,STATE2        : DRAM_STATE_TYPE;
SIGNAL STATE3              : CONTROL_STATE_TYPE;
SIGNAL STATE4              : ISA_STATE_TYPE;
SIGNAL IMAGEDETECT         : STD_LOGIC;
SIGNAL DRAM_STATE_START    : STD_LOGIC;
SIGNAL SYNCSTART           : STD_LOGIC;
SIGNAL SYNCIMAGE           : STD_LOGIC;
SIGNAL SET_IMAGECNT        : STD_LOGIC;
SIGNAL SET_DRAMHADDR       : STD_LOGIC;
SIGNAL SET_DRAMLADDR       : STD_LOGIC;
SIGNAL LOADDRAMHADDR       : STD_LOGIC;
SIGNAL LOADDRAMLADDR       : STD_LOGIC;
SIGNAL LOADIMAGECNT        : STD_LOGIC;
SIGNAL LOADLDONE           : STD_LOGIC;
SIGNAL LOADHDONE           : STD_LOGIC;
SIGNAL SYNCLOADDRAMHADDR   : STD_LOGIC;
SIGNAL SYNCLOADDRAMLADDR   : STD_LOGIC;
SIGNAL DRAMACCESRD         : STD_LOGIC;
SIGNAL CONFIGSELECT        : STD_LOGIC;
SIGNAL RAS1,CAS1           : STD_LOGIC;
SIGNAL RAS2,CAS2           : STD_LOGIC;
SIGNAL ADDRMUX             : STD_LOGIC;
SIGNAL RFADDR              : STD_LOGIC;
SIGNAL RF_TIMEOUT          : STD_LOGIC;
SIGNAL REQ_EVENT           : STD_LOGIC;
SIGNAL RESET               : STD_LOGIC;
SIGNAL TEMPSIGNAL          : STD_LOGIC;
SIGNAL TEMPSIGNAL2         : STD_LOGIC;
SIGNAL TEMPSTART           : STD_LOGIC;
SIGNAL TEMPDRAMLADDR       : STD_LOGIC;
SIGNAL TEMPDRAMHADDR       : STD_LOGIC;
SIGNAL TEMPSELECT          : STD_LOGIC;
SIGNAL TEMPFIRST1          : STD_LOGIC;
SIGNAL TEMPFIRST2          : STD_LOGIC;
SIGNAL TEMPFIRST3          : STD_LOGIC;
SIGNAL TEMPDRAMREADREQ     : STD_LOGIC;
SIGNAL STARTREADREQ        : STD_LOGIC;
```

```
SIGNAL DRAMREADREQ          : STD_LOGIC;
SIGNAL POLLING              : STD_LOGIC;
SIGNAL CONFIGMODE           : STD_LOGIC_VECTOR(1 DOWNTO 0);
SIGNAL RF_TIMEOUTCNT        : STD_LOGIC_VECTOR(6 DOWNTO 0);
SIGNAL IMAGECNT             : STD_LOGIC_VECTOR(5 DOWNTO 0);
SIGNAL N_SELECT             : STD_LOGIC_VECTOR(5 DOWNTO 0);
SIGNAL RFADDRCNT            : STD_LOGIC_VECTOR(10 DOWNTO 0);
-- ISA BUS Buffer : Used for the FPGA <-> ISA BUS interface
SIGNAL ISADATABUF           : STD_LOGIC_VECTOR(15 DOWNTO 0);
SIGNAL DRAMADDRCNT          : STD_LOGIC_VECTOR(21 DOWNTO 0);
-- DRAM DATA BUFFER : Used for the CAMERA -> DRAM interface
SIGNAL DRAMBUFFER           : STD_LOGIC_VECTOR(31 DOWNTO 0);
-- DRAM DATA BUFFER : Used for the ISA Bus <- DRAM interface
SIGNAL ISADRAMBUFFER        : STD_LOGIC_VECTOR(31 DOWNTO 0);


BEGIN
--*************************************************************
--* Process IO maps the Camera controller to ADDR 300-307H. *
--*************************************************************
 PROCESS(SA)
 BEGIN
  IF (SA(9 DOWNTO 3) = "1100000")AND(AEN = '0') THEN
     CASE SA(2 DOWNTO 0) IS
        WHEN "000" =>
             N_SELECT <= "111110";
        WHEN "001" =>
             N_SELECT <= "111101";
        WHEN "010" =>
             N_SELECT <= "111011";
        WHEN "011" =>
             N_SELECT <= "110111";
        WHEN "100" =>
             N_SELECT <= "101111";
        WHEN "110" =>
             N_SELECT <= "011111";
        WHEN OTHERS =>
             N_SELECT <= "111111";
     END CASE;
  ELSE
     N_SELECT <= "111111";
  END IF;
  -- Address 300H -> Configuration Select
  CONFIGSELECT    <= N_SELECT(0) OR N_IOWC;
  -- Address 301H -> Set Image Counter
  SET_IMAGECNT    <= N_SELECT(1) OR N_IOWC;
  -- Address 302H -> DRAM Acces
  DRAMACCESRD     <= N_SELECT(2) OR N_IORC;
  -- Address 303H -> Polling Port
  POLLING         <= N_SELECT(3) OR N_IORC;
  -- Address 304H -> Set DRAM Address Counter H Value
```

```
  SET_DRAMHADDR  <= N_SELECT(4) OR N_IOWC;
  -- Address 306H -> Set DRAM Address Counter L Value
  SET_DRAMLADDR  <= N_SELECT(5) OR N_IOWC;

 END PROCESS;

--*************************************************************************
--* Process is used to see what type of operation is going to be implemented *
--* To select operation write the following words to the CONFIGSELECT port   *
--* 00H : Default                                                            *
--* 01H : DRAM -> ISA Bus                                                    *
--* 02H : DRAM <- ISA Bus : Not Used                                         *
--* 03H : DRAM <- Camera                                                     *
--*************************************************************************
 PROCESS (N_IOWC,CONFIGSELECT)
 BEGIN
   IF N_IOWC'EVENT AND N_IOWC = '1' THEN
      IF CONFIGSELECT = '0' THEN
         CONFIGMODE <= PCDAT(1 DOWNTO 0);
      END IF;
   END IF;
 END PROCESS;

--*************************************************************************
--* Process is used to create a SYNCIMAGE signal that is 1 clock period  *
--* longer than the IMAGE signal.                                        *
--*************************************************************************
PROCESS(CONFIGMODE,ADCLK,IMAGE)
BEGIN
  IF ADCLK'EVENT AND ADCLK = '1' THEN
     SYNCIMAGE  <= IMAGE;
  END IF;
END PROCESS;

--*************************************************************************
--* Process is used to set the IMAGECNT and decrement the counter after  *
--* an IMAGE has been clocked into the DRAM.                             *
--*************************************************************************
PROCESS(CONFIGMODE,ADCLK,SYNCIMAGE)
BEGIN
  IF ADCLK'EVENT AND ADCLK = '1' THEN
     IF CONFIGMODE = "00" THEN
        IF LOADIMAGECNT = '1' THEN
           IMAGECNT <= ISADATABUF(5 DOWNTO 0);
        END IF;
     END IF;

     IF CONFIGMODE = "11" THEN
        IF (IMAGE = '0') AND (SYNCIMAGE = '1') AND (IMAGEDETECT = '1') THEN
           IF IMAGECNT /= 0 THEN
              IMAGECNT <= IMAGECNT - 1;
```

```
          END IF;
        END IF;
      END IF;
  END IF;
  END PROCESS;

--*************************************************************************
--* Process used to create IMAGEDETECT signal that will ensure that only *
--* full images will be clocked into the DRAM.                            *
--*************************************************************************
 PROCESS (CONFIGMODE,IMAGE,SYNCIMAGE,ADCLK)
 BEGIN
   IF CONFIGMODE /= "11" THEN
      IMAGEDETECT <= '0';
   ELSIF ADCLK'EVENT AND ADCLK = '1' THEN
         IF (IMAGE = '0') AND (SYNCIMAGE = '1') THEN
            IMAGEDETECT <= '1';
         END IF;
   END IF;
 END PROCESS;

--*************************************************************************
--* Process is used to set-up the DRAM address: 2 Low Bytes and 1 Hi Byte *
--* and the IMAGE Counter.                                                *
--*************************************************************************
PROCESS (N_IOWC,SET_DRAMHADDR,SET_DRAMLADDR,CONFIGMODE,SET_IMAGECNT,PCDAT)
BEGIN
  IF N_IOWC'EVENT AND N_IOWC = '1' THEN
     IF CONFIGMODE = "00" THEN
        IF SET_DRAMHADDR = '0' THEN
           LOADDRAMHADDR <= '1';
           ISADATABUF    <= PCDAT;
        ELSE
           LOADDRAMHADDR <= '0';
        END IF;

        IF SET_DRAMLADDR = '0' THEN
           LOADDRAMLADDR <= '1';
           ISADATABUF    <= PCDAT;
        ELSE
           LOADDRAMLADDR <= '0';
        END IF;

        IF SET_IMAGECNT = '0' THEN
           LOADIMAGECNT <= '1';
           ISADATABUF <= PCDAT;
        ELSE
           LOADIMAGECNT <= '0';
        END IF;
     ELSE
        LOADDRAMLADDR <= '0';
```

```
        LOADDRAMHADDR <= '0';
        LOADIMAGECNT  <= '0';
      END IF;

  END IF;
END PROCESS;

--*************************************************************************
--* Synchronise LOADDRAMLADDR and LOADDRAMHADDR with ADCLK using a       *
--* standard synchroniser.                                                *
--*************************************************************************
  PROCESS (LOADDRAMLADDR, LOADDRAMHADDR, ADCLK)
  BEGIN
    IF ADCLK'EVENT AND ADCLK = '0' THEN
       TEMPDRAMLADDR     <= LOADDRAMLADDR;
       TEMPDRAMHADDR     <= LOADDRAMHADDR;
    END IF;

    IF ADCLK'EVENT AND ADCLK = '0' THEN
       SYNCLOADDRAMLADDR <= TEMPDRAMLADDR;
       SYNCLOADDRAMHADDR <= TEMPDRAMHADDR;
    END IF;
  END PROCESS;


--*************************************************************************
--* Process creates an ADCLK sync signal to start the CAMERA State machine. *
--*************************************************************************
 PROCESS (ADCLK,CONFIGMODE)
 BEGIN
   IF ADCLK'EVENT AND ADCLK = '0' THEN
      IF CONFIGMODE  = "11" THEN
         TEMPSTART <= '1';
      ELSE
         TEMPSTART <= '0';
      END IF;
      SYNCSTART  <= TEMPSTART;
   END IF;
 END PROCESS;

--*************************************************************************
--* Process Informs the ISA Bus that a 16 Bit data transfer will occur.  *
--* The /IO16 line is pulled low when a valid ISA Bus Address and /IORC  *
--* or /IOWC is received. Otherwise the /IO16 line is tri-stated.        *
--*************************************************************************
 PROCESS (DRAMACCESRD,CONFIGSELECT,SET_DRAMHADDR,SET_DRAMLADDR,N_IORC,N_IOWC)
 BEGIN
   IF (DRAMACCESRD AND CONFIGSELECT AND
       SET_DRAMHADDR AND SET_DRAMLADDR) = '0' THEN
      N_IO16 <= '0';
   ELSE
```

```
      tADDRMUX := '0';
      tRFADDR  := '0';
      IF SYNCSTART = '1' THEN
         tN_DRWR := '0';
      ELSE
         tN_DRWR := '1';
      END IF;
    WHEN CAS =>
      tRAS     := '1';
      tCAS     := '1';
      tADDRMUX := '1';
      tRFADDR  := '1';
      tN_DRWR  := '1';
    WHEN RF =>
      tRAS     := '1';
      tCAS     := '1';
      tADDRMUX := '1';
      tRFADDR  := '0';
      tN_DRWR  := '1';
    WHEN RFCAS =>
      tRAS     := '1';
      tCAS     := '1';
      tADDRMUX := '1';
      tRFADDR  := '1';
      tN_DRWR  := '1';
    WHEN OTHERS =>
      tRAS     := '1';
      tCAS     := '1';
      tADDRMUX := '1';
      tN_DRWR  := '1';
  END CASE;

  IF ADCLK'EVENT AND ADCLK='1' THEN
    STATE   <= NEXTSTATE;
    RAS1    <= tRAS;
    CAS1    <= tCAS;
    RFADDR  <= tRFADDR;
    ADDRMUX <= tADDRMUX;
    N_DRWR  <= tN_DRWR;

    IF STATE = RF THEN
       RF_TIMEOUT <= '0';      -- After any RF the RF_TIMEOUT veriable is reset.
       RF_TIMEOUTCNT <= "0000000";
    ELSE
      IF RF_TIMEOUTCNT = 127 THEN
         IF ((SYNCSTART AND IMAGE) = '1') THEN
            RF_TIMEOUT <= '0';
         ELSE
            RF_TIMEOUT <= '1';
         END IF;
      ELSE
```

```
          RF_TIMEOUTCNT <= RF_TIMEOUTCNT + 1;
        END IF;
      END IF;
    END IF;
  END PROCESS;

--*************************************************************************
--* RAS2, CAS2 negative clk triggered.                                    *
--* CLK         : __|--|__|--|__|--|__|--|__|--|__|--|__|--               *
--* STATE_START : __|--------------------------------                     *
--* RAS2        : -----------|_____|-----|_____|--                       *
--* CAS2        : -----------------|_____|--------                       *
--*************************************************************************
 PROCESS(ADCLK,DRAM_STATE_START,SYNCLOADDRAMHADDR,SYNCLOADDRAMLADDR)
 VARIABLE NEXTSTATE2  : DRAM_STATE_TYPE;
 VARIABLE tRAS2,tCAS2 : STD_LOGIC;
 BEGIN
   CASE STATE2 IS
     WHEN IDLE =>
       IF DRAM_STATE_START = '1' THEN
         NEXTSTATE2 := RAS;
       ELSE
         IF RF_TIMEOUT = '1' THEN
            NEXTSTATE2 := RFCAS;
         ELSE
            NEXTSTATE2 := IDLE;
         END IF;
       END IF;
     WHEN RAS =>
         NEXTSTATE2 := CAS;
     WHEN CAS =>
         NEXTSTATE2 := RF;
     WHEN RF =>
         NEXTSTATE2 := IDLE;
     WHEN RFCAS =>
         NEXTSTATE2 := RF;
     WHEN OTHERS =>
         NEXTSTATE2 := IDLE;
   END CASE;

   CASE NEXTSTATE2 IS
     WHEN IDLE =>
       tRAS2 := '1';
       tCAS2 := '1';
     WHEN RAS =>
       tRAS2 := '0';
       tCAS2 := '1';
     WHEN CAS =>
       tRAS2 := '1';
       tCAS2 := '0';
     WHEN RF =>
```

```
        tRAS2 := '0';
        tCAS2 := '1';
      WHEN RFCAS =>
        tRAS2 := '1';
        tCAS2 := '0';
      WHEN OTHERS =>
        tRAS2 := '1';
        tCAS2 := '1';
    END CASE;

    IF ADCLK'EVENT AND ADCLK='0' THEN
      STATE2 <= NEXTSTATE2;
      RAS2 <= tRAS2;
      CAS2 <= tCAS2;

      IF STATE2 = RF THEN
         RFADDRCNT <= RFADDRCNT+1;
      END IF;

      LOADHDONE <= SYNCLOADDRAMHADDR;
      LOADLDONE <= SYNCLOADDRAMLADDR;

      IF (SYNCLOADDRAMHADDR = '1') AND (LOADHDONE = '0') THEN
         DRAMADDRCNT(21 DOWNTO 16) <= ISADATABUF(5 DOWNTO 0);
      ELSIF (SYNCLOADDRAMLADDR = '1') AND (LOADLDONE = '0') THEN
            DRAMADDRCNT(15 DOWNTO 0) <= ISADATABUF;
      ELSIF STATE2 = CAS THEN
            DRAMADDRCNT <= DRAMADDRCNT+1;
      END IF;
  END IF;
 END PROCESS;

--*************************************************************************
--* Process combines RAS & RAS2 to give N_RAS and CAS & CAS2 to give      *
--* N_CAS. It also multiplexes the Row, Col and RF Row Address of the DRAM. *
--*************************************************************************
--* CLK         : __|--|__|--|__|--|__|--|__|--|__|--|__                  *
--* STATE_START : __|-----------------------------------                  *
--* N_RAS       : -----------|________|--|_____|-----|__                  *
--* N_CAS       : -----------------|__|-----------------                  *
--*                 ::::::::::::::x:::::x:::::x::::::::::                *
--* DRAMADDR    :       ROW         COL   RFROW    ROW                    *
--*************************************************************************
 PROCESS(ADDRMUX,RFADDR)
 BEGIN
   N_RAS <= RAS1 AND RAS2;
   N_CAS <= CAS1 OR CAS2;
   IF RFADDR = '1' THEN
      DRADDR <= RFADDRCNT;
   ELSE
      IF ADDRMUX = '1' THEN
```

```
        DRADDR <= DRAMADDRCNT(9 DOWNTO 0);
      ELSE
        DRADDR <= DRAMADDRCNT(19 DOWNTO 10);
      END IF;
   END IF;

END PROCESS;

--************************************************************************
--* Process is used to Init. the DRAM State machine.                     *
--* The DRAM State machine is activated when data is read via the ISA    *
--* Bus from the DRAM or written to the DRAM.                            *
--************************************************************************
PROCESS (STATE3,DRAMREADREQ)
BEGIN
  IF (STATE3 = BYTE2) OR (DRAMREADREQ = '1') THEN
     DRAM_STATE_START <= '1';
  ELSE
     DRAM_STATE_START <= '0';
  END IF;
END PROCESS;

--************************************************************************
--* Process is used to create a glitchles REQ_EVENT that will clock      *
--* the STARTREADREQ D-FlipFlop.                                         *
--************************************************************************
PROCESS
(BCLK,TEMPFIRST1,TEMPFIRST2,TEMPFIRST3,TEMPSELECT,STATE4,CONFIGMODE,TEMPSIGNAL2)
BEGIN
  IF ((STATE4 = HI_WORD) AND (TEMPSELECT = '1')) OR
     (((TEMPFIRST1 OR TEMPFIRST2 OR TEMPFIRST3 ) = '1') AND (CONFIGMODE = "01")) THEN
     TEMPSIGNAL2 <= '1';
  ELSE
     TEMPSIGNAL2 <= '0';
  END IF;
  IF BCLK'EVENT AND BCLK = '1' THEN
     REQ_EVENT <= TEMPSIGNAL2;
  END IF;
END PROCESS;

--************************************************************************
--* Process is used to create the STARTREADREQ signal that will be       *
--* synchronised with the ADCLK to create the DRAMREADREQ signal.        *
--************************************************************************
PROCESS (REQ_EVENT,RESET)
BEGIN
  IF RESET = '1' THEN
     STARTREADREQ <= '0';
  ELSIF REQ_EVENT'EVENT AND REQ_EVENT = '1' THEN
     STARTREADREQ <= '1';
  END IF;
```

```
END PROCESS;

--*************************************************************************
--* Process is used to create an ADCLK synchronised signal DRAMREADREQ    *
--* that is used to Init. the DRAM State machine when data is transferred *
--* from the DRAM -> ISA Bus. DRAMREADREQ is only reset 1 ADCLK after     *
--* the DRAM State machine has started.                                   *
--*************************************************************************
PROCESS(ADCLK,STARTREADREQ,TEMPDRAMREADREQ,DRAMREADREQ,RESET,STATE2)
BEGIN
  IF RESET = '1' THEN
     TEMPDRAMREADREQ <= '0';
     DRAMREADREQ     <= '0';
  ELSIF ADCLK'EVENT AND ADCLK = '0' THEN
        TEMPDRAMREADREQ <= STARTREADREQ;
        DRAMREADREQ  <= TEMPDRAMREADREQ;
  END IF;
  IF ADCLK'EVENT AND ADCLK = '0' THEN
     IF STATE2 = RAS THEN
        RESET <= '1';
     ELSE
        RESET <= '0';
     END IF;
  END IF;
END PROCESS;

--*************************************************************************
--* Process creates an ISA State machine that controls the flow of data   *
--* from the DRAM -> ISA Bus.                                             *
--* Detailed functional description :                                     *
--* When the System is not configured for DRAM -> ISA the State machine   *
--* is placed in a default mode. This mode will cause the State machine   *
--* to init. a DRAM read cycle when the CONFIGMODE is for the first time  *
--* set to DRAM -> ISA. After this initial setting the DRAM read cycle    *
--* is only controlled by a DRAMACCESRD via the ISA Bus.                  *
--*************************************************************************
PROCESS(BCLK,DRAMACCESRD,TEMPFIRST1,TEMPFIRST2,CONFIGMODE,TEMPSIGNAL)
VARIABLE NEXTSTATE4 : ISA_STATE_TYPE;
BEGIN
  CASE STATE4 IS
  WHEN LOW_WORD =>
       NEXTSTATE4 := HI_WORD;
  WHEN HI_WORD  =>
       NEXTSTATE4 := LOW_WORD;
  WHEN OTHERS   =>
       NEXTSTATE4 := LOW_WORD;
  END CASE;

  IF DRAMACCESRD = '0' THEN
     TEMPSELECT  <= '1';
     TEMPSIGNAL  <= '1';
```

```
  ELSIF BCLK'EVENT AND BCLK = '1' THEN
     IF DRAMACCESRD = '1' THEN
        TEMPSIGNAL <= '0';
     END IF;
     TEMPSELECT  <= TEMPSIGNAL;
  END IF;

  IF CONFIGMODE /= "01" THEN
     STATE4 <= LOW_WORD;
     TEMPFIRST1 <= '1';
     TEMPFIRST2 <= '1';
     TEMPFIRST3 <= '1';
  ELSIF BCLK'EVENT AND BCLK = '1' THEN
        IF (TEMPSELECT = '1') AND (DRAMACCESRD = '1') AND
           (TEMPSIGNAL = '0') THEN
           STATE4 <= NEXTSTATE4;
        END IF;
        TEMPFIRST1 <= '0';
        TEMPFIRST2 <= TEMPFIRST1;
        TEMPFIRST3 <= TEMPFIRST2;
  END IF;
END PROCESS;

--*************************************************************************
--* Process is used to control the flow of data from the Camera to the  *
--* DRAMBUFFER.                                                         *
--*************************************************************************
 PROCESS(SYNCSTART,IMAGE,ADCLK)
 VARIABLE NEXTSTATE3 : CONTROL_STATE_TYPE;
 BEGIN
   CASE STATE3 IS
   WHEN WAITSTATE =>
        IF ((SYNCSTART AND IMAGE) = '1') AND (IMAGECNT /= 0) AND (IMAGEDETECT = '1')
THEN
           NEXTSTATE3 := BYTE0;
        ELSE
           NEXTSTATE3 := WAITSTATE;
        END IF;
   WHEN BYTE0 =>
        NEXTSTATE3 := BYTE1;
   WHEN BYTE1 =>
        NEXTSTATE3 := BYTE2;
   WHEN BYTE2 =>
        NEXTSTATE3 := BYTE3;
   WHEN BYTE3 =>
        IF (SYNCSTART AND IMAGE) = '1' THEN
           NEXTSTATE3 := BYTE0;
        ELSE
           NEXTSTATE3 := WAITSTATE;
        END IF;
   WHEN OTHERS =>
```

```
        NEXTSTATE3 := WAITSTATE;
   END CASE;

   IF ADCLK'EVENT AND ADCLK = '0' THEN
      STATE3 <= NEXTSTATE3;
   END IF;

   IF ADCLK'EVENT AND ADCLK = '0' THEN
     IF STATE3 = BYTE0 THEN
       DRAMBUFFER(7 DOWNTO 0)   <= CAMDAT;
     END IF;
     IF STATE3 = BYTE1 THEN
       DRAMBUFFER(15 DOWNTO 8)  <= CAMDAT;
     END IF;
     IF STATE3 = BYTE2 THEN
       DRAMBUFFER(23 DOWNTO 16) <= CAMDAT;
     END IF;
     IF STATE3 = BYTE3 THEN
       DRAMBUFFER(31 DOWNTO 24) <= CAMDAT;
     END IF;
   END IF;
 END PROCESS;

--************************************************************************
--* Process performs the reading and writing of data to and from the    *
--* DRAM.                                                                *
--************************************************************************
 PROCESS(ADCLK,SYNCSTART,DRAMBUFFER,STATE)
 BEGIN
   IF SYNCSTART = '0' THEN
      DRAM <= "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ";
   ELSE
      DRAM <= DRAMBUFFER;
   END IF;

   IF ADCLK'EVENT AND ADCLK = '1' THEN
      IF STATE = RAS THEN
         ISADRAMBUFFER <= DRAM;
      END IF;
   END IF;
 END PROCESS;

--************************************************************************
--* Process performs the writing of DRAM data to the ISA Bus.           *
--************************************************************************
 PROCESS(DRAMACCESRD,STATE4,CONFIGSELECT,POLLING)
 BEGIN
   IF (DRAMACCESRD AND POLLING) = '1'  THEN
      DIR <= '1';
      PCDAT <= "ZZZZZZZZZZZZZZZZ";
   ELSE
```

```
      DIR <= '0';
      IF (IMAGECNT = 0) AND (POLLING = '0')  THEN
            PCDAT <= "0000000000000001";
      ELSIF (IMAGECNT /= 0) AND (POLLING = '0') THEN
            PCDAT <= "0000000000000000";
      ELSE
        IF STATE4 = LOW_WORD THEN
           PCDAT <= ISADRAMBUFFER(15 DOWNTO 0);
        ELSE
           PCDAT <= ISADRAMBUFFER(31 DOWNTO 16);
        END IF;
      END IF;
    END IF;
  END PROCESS;


  --#########################################################################
  NOWS   <= 'Z';
  N_BEN  <= '0'; --Bus drivers are always active.

END CAMISA;
```

APPENDIX G

SCHEMATIC FOR THE MVCI CARD.

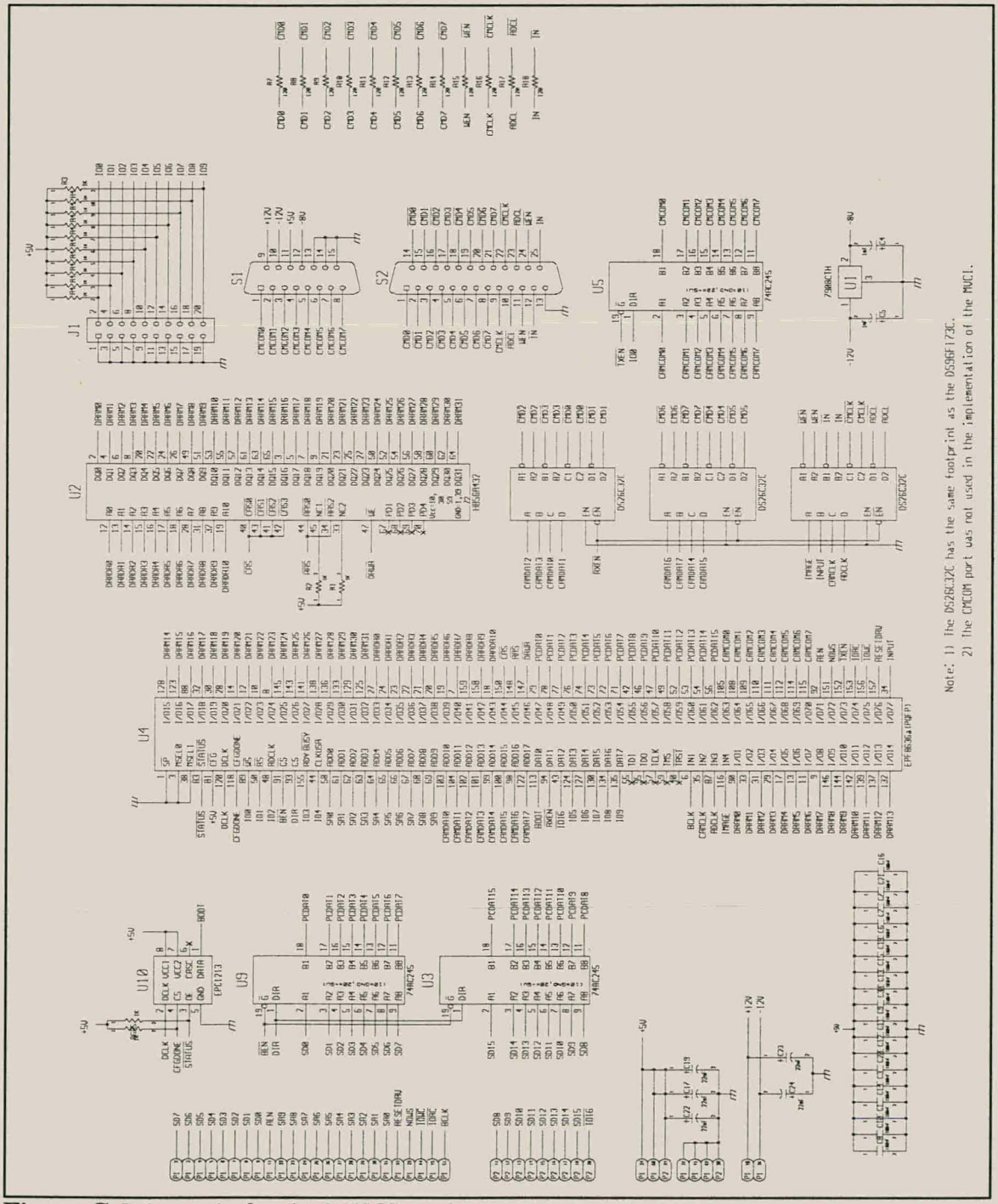

Figure G Schematic for the MVCI card.

Note: For the current implementation of the MVCI the MVC command port S1 is not used.

APPENDIX H

EXAMPLES OF RGB COLOUR IMAGES PRODUCED BY THE MVC.

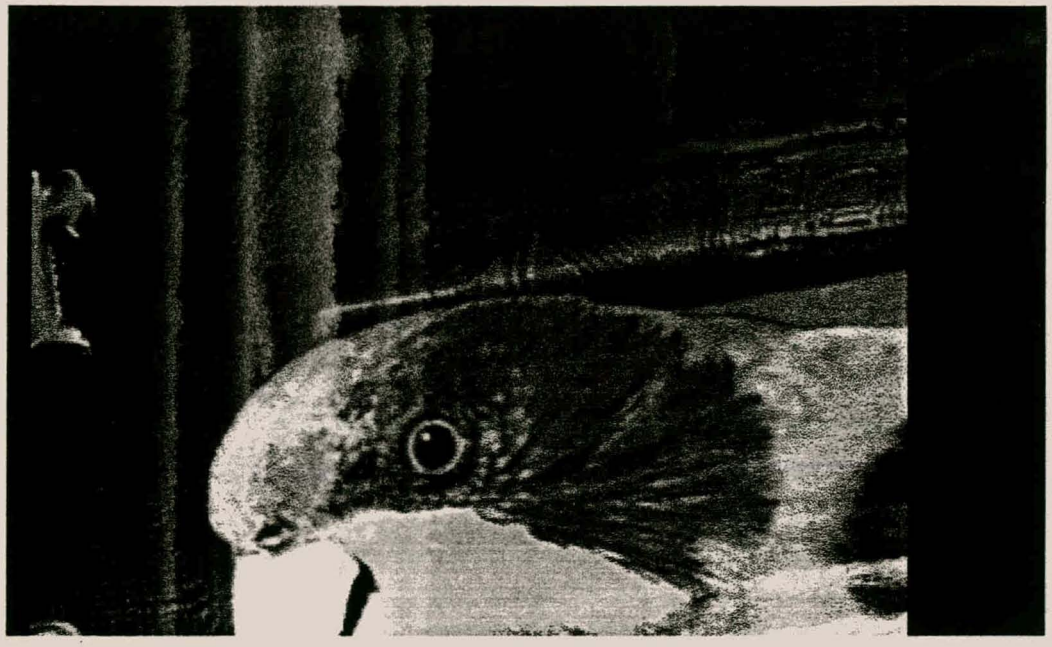

Figure H.1 Image of a picture captured from a distance closer than 0.5m.

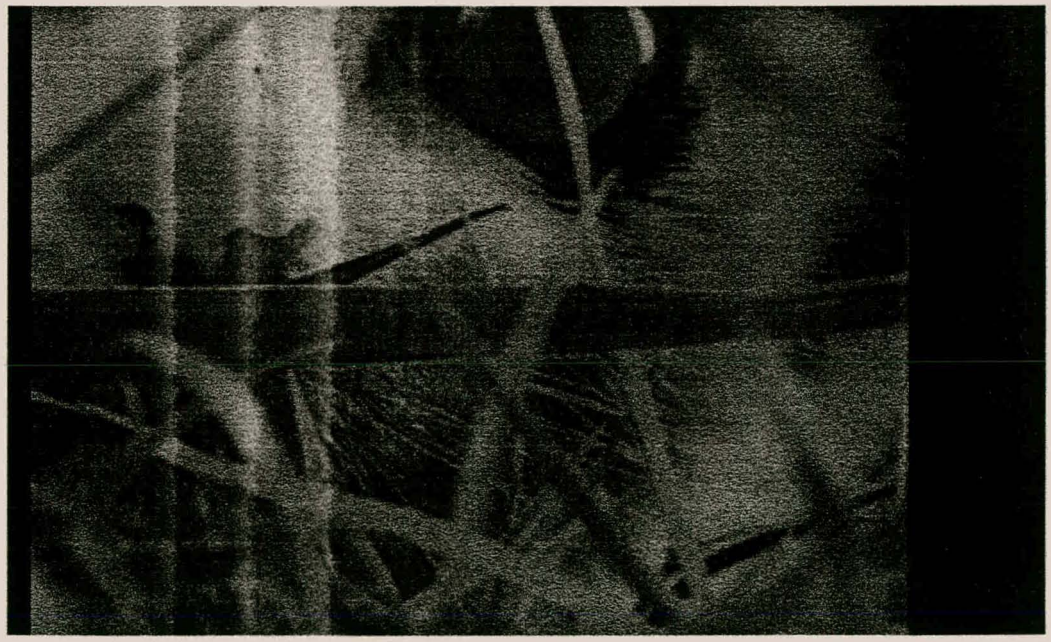

Figure H.2 Image of a mouse pad captured from a distance closer than 0.5m.

Figure H.3 Image of a photo captured from a distance closer than 0.5m.

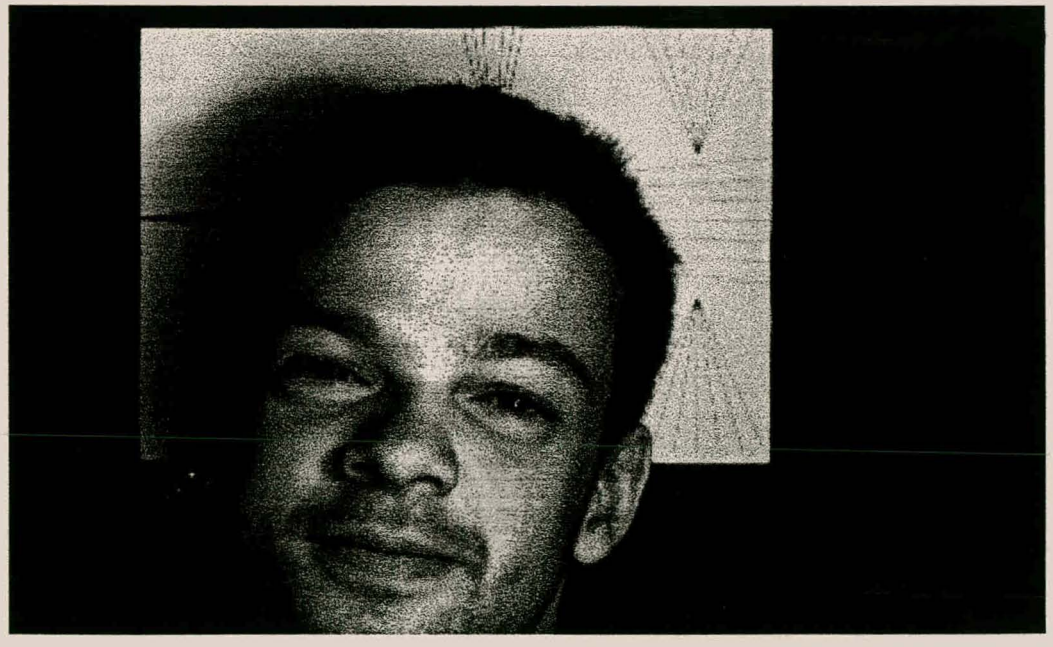

Figure H.4 Image of a person captured from a distance of 2m.

Figure H.5 Image of a poster captured from a distance of 2m.

Figure H.6 Image of another poster captured from a distance of 2m.

APPENDIX I

ROWLEY MODULA-2 SOURCE CODE FOR THE 32-BIT RADIX-2 DIT 1D-FFT FOR THE T800 TRANSPUTER.

The structure of the algorithm is similar to the one implemented by Brigham (1988:131).

```
MODULE FFT;

(*$ code     (cpu => t8)          *)(* Compile for T800 transputer      *)
(*$ data     (align => on)        *)(* Data aligned with word addresses *)
(*$ optimize(all => on)           *)(* Active all optimisations         *)
(*$ optimize(strategy => speed)*)(* Optimise additionally for speed  *)


FROM BlockOps      IMPORT BlkFill;
FROM Terminal      IMPORT ReadLn;
FROM InOut         IMPORT WriteString,WriteLn,WriteCard;
FROM RealInOut     IMPORT WriteReal;
FROM MathLib0      IMPORT pi,ln,sin,cos,power;
FROM SYSTEM        IMPORT ADR;
FROM Timer         IMPORT Now;

TYPE
  DataType      = ARRAY[0..1023] OF REAL;
  Lookuparray   = ARRAY[0..10]   OF REAL;
  IBRarray      = ARRAY[0..1023] OF CARDINAL;
  SCarray       = ARRAY[0..1023] OF REAL;

VAR
  COS,SIN      : SCarray;      (* Twiddle factor lookup table *)
  LkUpIBR      : IBRarray;     (* Bit-reversal lookup table   *)
  LkUpPower    : Lookuparray;  (* 2^power lookup table        *)
  rx,ix        : DataType;     (* real and complex data array *)
  Loop         : CARDINAL;     (* Loop counter *)
  Time1,Time2  : INTEGER;      (* Time variables              *)
  N            : CARDINAL;     (* Transform length            *)

(*-------------------------------------------------------------------*)

PROCEDURE IBR(k,NU : CARDINAL) : CARDINAL;

(* Function calculates the bit-reversed version of the NU-bit number k.
*)

VAR
 i,m,a,t,mt    : CARDINAL;

BEGIN
  m  := k;
  mt := 0;
  FOR i := 1 TO NU DO
      a  := m DIV 2;
```

```
      mt := mt*2+(m-2*a);
      m  := a;
  END; (*IF*)
  RETURN mt;
END IBR;

(*-------------------------------------------------------------------*)

PROCEDURE IBR_Look_Up(VAR LkUpIBR: IBRarray; N : CARDINAL);

(* Procedure calculates a lookup table for the Twiddle factors (W^p)
   power value (p). *)

VAR
  tNU   : REAL;
  k,NU  : CARDINAL;

BEGIN
  tNU := ln(FLOAT(N))/ln(2.0);
  NU  := TRUNC(tNU);
  FOR k := 0 TO (N-1) DO
      LkUpIBR[k] := IBR(k,NU);
  END; (*FOR*)
END IBR_Look_Up;

(*-------------------------------------------------------------------*)

PROCEDURE Look_Up_Power(VAR LkUpPower : Lookuparray; N : CARDINAL);

(* Procedure calculates a lookup table for 2^i *)

VAR
  tNU   : REAL;
  i,NU  : CARDINAL;

BEGIN
  tNU := ln(FLOAT(N))/ln(2.0);
  NU  := TRUNC(tNU);
  FOR i := 0 TO NU DO
      LkUpPower[i] := power(2.0,FLOAT(i));
  END; (*FOR*)
END Look_Up_Power;
(*-------------------------------------------------------------------*)

PROCEDURE SinCos_Lookup(VAR SIN,COS : SCarray; N : CARDINAL);

(* Procedure calculates a lookup table for the real and imaginary
   components of the Twiddle factors W^p *)

VAR
  i : CARDINAL;

BEGIN
  FOR i := 0 TO (N-1) DO
      COS[i] := cos(-2.0*pi/FLOAT(N)*FLOAT(i));
      SIN[i] := sin(-2.0*pi/FLOAT(N)*FLOAT(i));
  END; (*FOR*)
```

```
END SinCos_Lookup;

(*-------------------------------------------------------------------*)

PROCEDURE STD_FFT(VAR rx,ix : DataType; N : CARDINAL);

(* Procedure calculates a N-point 1D-FFT *)

VAR
  N2,    (* Represent the spacing between dual nodes                *)
  T1,    (* Index for the bit-reversal lookup table                 *)
  l,     (* Represents the current stage of the 1D-FFT calculation  *)
  i,     (* Loop counter                                            *)
  k,     (* Loop counter                                            *)
  NU,    (* Number of stages in the 1D-FFT calculation              *)
  p      (* Rxponential function W's power value                    *)
         : CARDINAL;
  tT1,   (* Temp. variables                                         *)
  tN2,
  tNU,
  rXt,
  iXt,
  rTemp,
  iTemp  : REAL;

BEGIN
  tNU := ln(FLOAT(N))/ln(2.0);
  NU  := TRUNC(tNU);

  FOR l := 1 TO NU DO
      k  := 0;
      i  := 0;
      tN2 := FLOAT(N)/LkUpPower[l];
      N2  := TRUNC(tN2);

      WHILE (k < N-1) DO

            tT1 := FLOAT(k)/LkUpPower[NU-l];
            T1  := TRUNC(tT1);
            p   := LkUpIBR[T1];

            IF i < N2 THEN
               rTemp := COS[p]*rx[k+N2]-SIN[p]*ix[k+N2];
               iTemp := COS[p]*ix[k+N2]+SIN[p]*rx[k+N2];
               (* DIT FFT butterfly calculation *)
               rx[k+N2] := rx[k]-rTemp;
               ix[k+N2] := ix[k]-iTemp;
               rx[k]    := rx[k]+rTemp;
               ix[k]    := ix[k]+iTemp;
               i := i+1;
               k := k+1;
            ELSE
               k := k+N2; (* Skip dual nodes *)
               i := 0;

            END; (*IF i*)
      END; (*WHILE*)
```

```
  END; (* FOR 1*)

  FOR k := 0 TO N-1 DO

     (* Shuffle 1D-FFT result back into normal order *)
      i := LkUpIBR[k];
      IF i > k THEN
         rXt   := rx[k];
         iXt   := ix[k];
         rx[k] := rx[i];
         ix[k] := ix[i];
         rx[i] := rXt;
         ix[i] := iXt;
      END (*IF i*)
  END; (*FOR k*);

END STD_FFT;

(*=========================================================================*)

BEGIN

  WriteString('Program started'); WriteLn;

  BlkFill(ADR(rx),SIZE(rx),CARD_1(0));(* Fill real array with 0      *)
  BlkFill(ADR(ix),SIZE(ix),CARD_1(0));(* Fill imaginary array with 0 *)

  (* Fill real array with test data *)

  rx[0]  :=  1.0;
  rx[1]  :=  3.0;
  rx[2]  :=  5.0;
  rx[3]  :=  7.0;
  rx[4]  :=  9.0;
  rx[5]  := -11.0;
  rx[6]  :=  12.0;
  rx[7]  := -13.0;
  rx[8]  :=  45.0;
  rx[9]  :=  20.0;
  rx[10] := -22.0;
  rx[11] := -45.0;
  rx[12] :=  11.0;
  rx[13] :=  8.0;
  rx[14] :=  0.0;
  rx[15] :=  20.0;

  N := 1024; (* Calculate a 1024-point FFT *)

  Look_Up_Power(LkUpPower,N); (* Create all lookup tables *)
  IBR_Look_Up(LkUpIBR,N);
  SinCos_Lookup(SIN,COS,N);

  Time1 := Now();              (* Get first timer value  *)
  STD_FFT(rx,ix,N);            (* Start FFT calculation  *)
  Time2 := Now();              (* Get second timer value *)

  WriteString('Time => '); WriteReal(FLOAT(Time2-Time1)/15625.0,15);
```

```
  WriteLn;

  WriteString(' R                I'); WriteLn;
  FOR Loop := 0 TO 15 DO
      WriteReal(rx[Loop],15); WriteString(' '); WriteReal(ix[Loop],15);
      WriteLn;
  END; (*FOR*)

END FFT.
```

NOTE : The source code for the 64-bit Radix-2 DIT 1D-FFT is the same as the 32-bit version, except that all real variables are replaced by long real variables and all real functions are replaced by their respective long real function.

APPENDIX J

ROWLEY MODULA-2 SOURCE CODE FOR THE T800 TRANSPUTER AND FPGA DATA TRANSFER.

```
MODULE scale;

(*$ code     (cpu => t8)          *) (* Compile for T800 transputer      *)
(*$ data     (align => on)        *) (* Data aligned with word addresses *)
(*$ optimize(all => on)           *) (* Active all optimisations         *)
(*$ optimize(strategy => speed)   *) (* Optimise additionally for speed  *)

FROM SYSTEM     IMPORT ADR;
FROM InOut      IMPORT WriteString,WriteLn,WriteInt,WriteCard,WriteHex;
FROM RealInOut  IMPORT WriteReal;
FROM Timer      IMPORT Now;
FROM BlockOps   IMPORT BlkFill,BlkMove;
FROM MathLib0   IMPORT entier;
FROM SYSTEM     IMPORT BAND;

CONST

  NUMBER = 1024; (* Number of complex valued data to transfer to and from the
                    FPGA. *)

TYPE
   INT_ARR  = ARRAY[0..NUMBER]  OF INT_4;
   REAL_ARR = ARRAY[0..NUMBER]  OF REAL;


VAR
  X,
  Y          : CARD_4;    (* Loop counter *)

  R_NUM,
  I_NUM,
  I_REAL,
  R_REAL,
  R_Scaler,
  I_Scaler   : REAL;      (* Temp. variables *)

  R_TEMP,
  I_TEMP,
  COUNT1,
  COUNT2     : INT_4;     (* Temp. variables *)

  FPGA       : INT_ARR;   (* Array containg data to and from FPGA *)

  R_DAT,
  I_DAT      : REAL_ARR;  (* Arrays containing the real and imaginary data to be
                             FFTed *)

(*-------------------------------------------------------------------------*)
```

```
PROCEDURE Max_Num (VAR DATA : REAL_ARR) : REAL;

(* Function calculates the max. abs. value in the array DATA *)

VAR
  NUM : REAL;
  N   : CARD_4;

BEGIN
  NUM := 0.0;

  FOR N := 0 TO NUMBER DO
    IF NUM <  ABS(DATA[N]) THEN
       NUM := ABS(DATA[N]);
    END;
  END;

  RETURN (NUM);

END Max_Num;

(*=============================================================================*)

BEGIN
  WriteString('Program Started 1');
  WriteLn;

  BlkFill(ADR(R_DAT),NUMBER*4,CARD_1(0)); (* Fill real array with 0      *)
  BlkFill(ADR(I_DAT),NUMBER*4,CARD_1(0)); (* Fill imaginary array with 0 *)

  (* Fill imaginary array with test data *)

  I_DAT[0] := 1.5;
  I_DAT[1] := 2.2;
  I_DAT[2] :=-3.123;
  I_DAT[3] := 4.45;
  I_DAT[4] :=-5.6;

  (* Fill real array with test data *)

  R_DAT[0] := 1.0;
  R_DAT[1] := 2.0;
  R_DAT[2] :=-3.0;
  R_DAT[3] := 4.0;
  R_DAT[4] :=-5.0;


  COUNT1 := Now();             (* Get first timer value             *)

   FOR Y := 1 TO 1000 DO       (* Test is repeat a 1000 times       *)

     I_NUM := Max_Num(I_DAT); (* Find the max. abs. value in array *)
     R_NUM := Max_Num(R_DAT);
```

```
    (* Calculate scalar to convert data between -1 and 1 *)

    R_Scaler := 32767.0/R_NUM;
    I_Scaler := 32767.0/I_NUM;


    (* Convert 32-bit complex valued data to 16-bit fixed-point data and ouput
      to FPFA. *)

    FOR X := 0 TO NUMBER DO

      R_TEMP := entier(R_DAT[X]*R_Scaler);  (* Scale data to 16-bit rep.  *)
      I_TEMP := entier(I_DAT[X]*I_Scaler);  (* Scale data to 16-bit rep.  *)

      IF R_TEMP < 0 THEN
         R_TEMP := R_TEMP + 65536;          (* 2's complement notation    *)
      END;

      IF I_TEMP < 0 THEN
         I_TEMP := I_TEMP + 65536;
      END;

      FPGA[X] := R_TEMP+65536*I_TEMP;       (* Output INT_4 data to FPGA  *)

    END;

    (* Convert 16-bit fixed-point data from FPGA to 32-bit complex valued data
       for the T800 transputer.*)

    FOR X := 0 TO NUMBER DO

     (* Split FPGA data back into real and imaginary components *)

      I_REAL := FLOAT(FPGA[X])/65536.0;
      R_REAL := FLOAT(BAND(FPGA[X],INT_4(0000FFFFH)));

      IF R_REAL > 32767.0 THEN
         R_REAL := R_REAL - 65536.0;    (* Inverse 2's complement notation *)
      END;

      IF I_REAL > 32767.0 THEN
         I_REAL := I_REAL - 65536.0;
      END;

      I_DAT[X] := I_REAL/I_Scaler;      (* Scale data back to 32-bit reals *)
      R_DAT[X] := R_REAL/R_Scaler;

      (* NOTE : The 2^exponent is left out, since it will not have a significant
         impact of the overall calculations *)

    END;

END;

COUNT2 := Now();                        (* Get second timer value          *)
```

```
  WriteString('Program End');

  WriteLn;
  WriteInt(COUNT2-COUNT1,10);
  WriteLn;

  WriteCard(FPGA[0],15);
  WriteLn;
  WriteCard(FPGA[1],15);
  WriteLn;
  WriteCard(FPGA[2],15);
  WriteLn;
  WriteCard(FPGA[3],15);
  WriteLn;
  WriteCard(FPGA[4],15);

  WriteLn;
  WriteReal(R_DAT[0],15);
  WriteLn;
  WriteReal(R_DAT[1],15);
  WriteLn;
  WriteReal(R_DAT[2],15);
  WriteLn;
  WriteReal(R_DAT[3],15);
  WriteLn;
  WriteReal(R_DAT[4],15);


END scale.
```

APPENDIX K

MATLAB SOURCE CODE FOR THE DICC FUNCTION.

The following Matlab routine implements the DICC function defined in Chapter 6.

```
clear all;
format compact;


% Implementing xcorr2(IMAGE1,IMAGE2) through the use of fft2.

R = 9;
C = 10;

tic % Set stopwatch

A = fft2(IMAGE1,2^R,2^C); %Zero pad image to 1024 (H) x 512 (V) and
                          calculate the 2D-FFT of the image.

B = fft2(ROT90(ROT90(IMAGE2)),2^R,2^C); %Zero pad image to 1024 (H) x
512 (V); rotate IMAGE2 through 180 deg. and calculate the 2D-FFT.

C = (A.*B);               %Compute correlation in frequancy domain.

[Rc,Cc] = size(C);

D = 1/(Rc*Cc)*real(fft2(conj(C))); % Take complex conjucute of C to
implement and implement the 2D-IFFT2 through the use of the 2D-FFT2.

toc % Stop stopwatch and display elapsed time
```

REFERENCES

ALTERA, (1996) "*ALTERA 1996 Data Book: The ALTERA Advantage*" ALTERA Corporation: pp.193-261.

ASPRS, (American Society for Photogrammetry and Remote Sensing) (1994) "*Digital Photogrammetry: An Addendum to the Manual of Photogrammetry*"

Atmel, (1998) "*AT40K-FFT (AT40K FPGA IP Core)*" Rev. 1132A

Becke G., Forstner P., Haseloff E., and Huchzermeier J., (1997) "*Digital Design Seminar Reference manual*" Texas Instruments, pp.3-11.

Brigham E.O., (1988) "*The Fast Fourier Transform and its Applications.*" Prentice Hall signal processing series.

Buil C., (1991) "*CCD Astronomy*" Willman-Bell Incorporated

Burt D.J., (s.d.) "*CCD Performance Limitations: Theory and Practice.*" EEV Ltd., CCD Technical Note 18.

EEV, (1987) "*CCD IMAGING III.*" EEV Ltd.

Fleury M., and Clark A.F., (1998) "Parallelising a set of 2D frequency transforms in a flexible manner" in *"IEE Proc. –Vis., Image Signal Processing,"* 145(1):65-72.

HITACHI, (1994) "*HM5117400A/Al Series 4194304-word x 4-bit Dynamic Random Access Memory*" Hitachi semiconductor, Rev. 3.0.

INMOS, (1985) "*IMS B004 evaluation board User manual*" INMOS

Jacobs M.L., (1995) "*A low cost high precision star sensor*" Masters Thesis, University of Stellenbosch.

Jain A.K., (1988) "*Fundamentals of Digital Image Processing*" Prentice-Hall International.

Moolman, D.W., (1995a) *"The interpretation of flotation froths by using computer vision and connectionist networks."* PhD. Thesis, University of Stellenbosch.

Moolman D.W., Aldrich C., van Deventer J.S.J., and Brandshaw D.J., (1995b) "The interpretation of flotation froth surfaces by using digital image analysis and neural networks" in *"Chemical Engineering Science"* 50(22):3501-3513.

Moolman D.W., Eksteen J.J., Aldrich C., and van Deventer J.S.J., (1996) "The significance of flotation froth appearance for machine vision control" in *"International Journal of Mineral Processing"* 48(3-4):135-158.

Proakis J.G., and Manolakis D.G., (1996) "*Digital Signal Processing: Principles, Algorithms, and Applications*" Third Edition, Prentice Hall.

RCA, (1974) "*Electro-Optics Handbook*" Technical series, E.O.H., Second Edition.

Roebbers H., (1990) "*A generalized FFT algorithm on transputers*" Transputer Research and Applications 4, IOS Press, pp.77-87.

Rowley, (1992) "*Rowley Modula-2*" Rowley Associates, Third edition, Version 1.4.8.

SGS-THOMSON, (1989) "*INMOS TRANSPUTER DATABOOK*" Second Edition.

Siliconix, (1982) "*Small Signal FET Design Catalog*" Siliconix, ch.2, p.29.

SONY, (1997a) "*CCD Area Image Sensor Data Book*" SONY Semiconductors.

SONY, (1997b) "*SONY Semiconductor Data Book*" SONY Semiconductors, Compact disc for Windows.

Sun. C., and Wee W.G., (1982) "Neighbouring Grey Level Dependence Matrix for Texture Classification" in "*Computer vision, Graphics and Image Processing*," 23:341-352.

Wills, B.A., (1981) *"Mineral Processing Technology: An Introduction to the practical Aspects of Ore Treatment and Mineral Recovery."* Second Edition, Pergamon Press, Oxford.

Xilinx, (1998) "*CORE Solutions Databook*" Xilinx, ch.2, p.69.

ELECTRONIC REFERENCES

ALTERA, (1997) "*Fast Fourier Transform Data Sheet*" ver. 3, www.altera.com.

ALTERA, ftp://ftp.altera.com/pub/megacore/fft/

DALSA, (1997a) "*Area Scan Primer and Selection Guide*", http://www.dalsa.com

DALSA, (1997b) "*Understanding Sensor Noise Specifications*" Application notes, http://www.dalsa.com

DALSA, (1997c) "*Sensor Measurement Definitions*" Application notes, www.dalsa.com

DALSA, (1997d) "*Measurement Definitions and Techniques*" Camera Application Notes, www.dalsa.com

Edmund Scientific, (1998a) "*Video & Electronic Imaging Primer*" Technical Library, http://www.edsci.com/industrial/tect_lib/video_primers/color_monochrome.html

Edmund Scientific, (1998b) "*Video & Electronic Imaging Primer*" Technical Library, http://www.edsci.com/industrial/tect_lib/video_primers/color_video.html

National, (1996) "*DS96F172/DS96F172M/DS96F174C/DS96F174M EIA-485/EIA-422 Quad Differential Drivers*" National Semiconductors, http://www.national.com.

Nelson T., (1995) "*The Practical Limits of RS-485*" National Semiconductor, Application Note 979, http://www.national.com.

Photobit, (1998a) "*Technology : CMOS Active Pixel Sensor Technology*" Photobit, http://www.photobit.com/tech.html.

Photobit, (1998b) "*Advantages Over the CCD*" Photobit, http://www.photobit.com