

Development of High-Speed CCD Control System for Adaptive Optics

Ee-Eul Kim

A Thesis Submitted for the Degree of
Doctor of Philosophy of the University of London



**Department of Physics and Astronomy
University College London
University of London
1998**

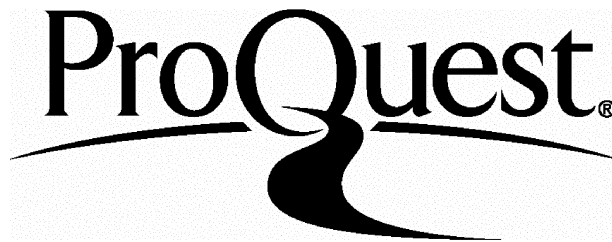
ProQuest Number: U642133

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest U642133

Published by ProQuest LLC(2015). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code.
Microform Edition © ProQuest LLC.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

The air turbulence in the Earth's atmosphere causes the wavefront distortion of the incoming light from an object and the degradation of the image quality for a ground-based astronomical telescope. The adaptive optics is the technique that corrects the wavefront distortion and attempts to restore the image to a quality close to the diffraction limit in real-time. The wavefront sensor, one of the fundamental elements in an adaptive optics system, is used to measure the wavefront distortion. The charge-coupled device(CCD) is the detector that is most widely employed for wavefront sensors because of its high quantum efficiency, 100 % fill-factor, and precise pixel positions.

This thesis describes the design and construction of a high-speed CCD control system primarily for wavefront sensors. The control system employs a high-speed FPGA(field-programmable gate array) as the sequencer and high-speed CCD clock drivers. It has the versatility to generate streams of clock and control pulses and biases for various different types of high-speed CCDs. It incorporates the modular graphical user-interface that provides a graphical and easy-to-use control environment to the system user. With the host personal computer(PC) and the main power supply unit, the control system is compact, modular, portable, and stand-alone. These features make the control system satisfy the required spatial resolution and temporal sampling frequency for wavefront sensing in different environments. Its versatility and modularity are also instrumental in obtaining the optimum performance of the control system for a specific condition. The results from some primary tests performed on the control system are presented and discussed. With some more tests and optimisation, the control system will be applicable to practical wavefront sensing for ground-based astronomical telescopes. Moreover, with little physical modification, it can be applied to numerous fields of science and engineering.

To my parents
Without whom I am nothing

Table of Contents

<i>Abstract</i>	<i>2</i>
<i>Table of Contents</i>	<i>4</i>
<i>List of Tables</i>	<i>8</i>
<i>List of Figures</i>	<i>9</i>
<i>1 Introduction</i>	<i>11</i>
1.1 Adaptive Optics	11
1.2 Wavefront Sensor	14
1.3 High-Speed CCD Control System for Adaptive Optics	16
1.4 Contents of the Thesis and Author's Contribution	18
<i>2 Charge-Coupled Device</i>	<i>23</i>
2.1 Characteristics of the CCD	25
2.1.1 Quantum Efficiency	25
2.1.2 Readout Noise	27
2.1.3 Charge Transfer Efficiency	28
2.1.4 Physical Size	29
2.1.5 CCDs in X-ray	29
2.1.6 CCDs in Infrared	30
2.2 CCDs for High-Speed Applications	32
<i>3 High-Speed CCD Control System for Adaptive Optics</i>	<i>34</i>
3.1 Wavefront Sensors for Astronomical Telescopes	37
3.2 Design Considerations	39
3.2.1 Readout Frequency	39
3.2.2 Versatile Control System	41
3.2.2.1 Bitmaps as CCD Clock and Control Timing Pulses	42

3.2.2.2	Static DAC Control	46
3.2.3	Modular and Compact System with the User-Friendly Interface	47
3.3	Other High-Speed CCD Imaging Systems	49
3.3.1	Motion Analysis and High-Speed Image Acquisition	49
3.3.2	High-Speed Photometry	51
3.3.3	Soft X-ray Imaging	52
4	<i>Prototype Control System Hardware</i>	54
4.1	Loral 64 × 64 Frame-Transfer CCD	57
4.2	Host PC	58
4.3	Main Power Supply Unit and Voltage Regulation Board	60
4.4	Bitmap Generation Board	60
4.4.1	Overview	60
4.4.2	Sequencer, EPM7256E-12	62
4.4.2.1	Sequencer Logic	63
4.4.2.2	Interface with the Host PC and DAC Control	67
4.4.2.3	Bitmap RAMs and Their Mapping	67
4.4.2.4	Writing of System Parameters and Bitmaps	68
4.4.2.5	Readout of the CCD	69
4.4.2.6	Miscellaneous Functions	71
4.4.3	Other Components	73
4.5	Bias Board	73
4.5.1	Overview	73
4.5.2	Octal DAC for CCD Biases	74
4.5.3	Translation and Amplification of the DAC Output	75
4.6	Clock Boards	76
4.6.1	Overview	76
4.6.2	Octal DACs with the Static DAC Control Method	77
4.6.3	Generation of CCD Clock Pulse Streams	78
4.7	Signal Processing Boards	81
4.7.1	Overview	81
4.7.2	Video Signal Processing	82
4.7.3	Control of Video Signal Sampling and ADC Conversion	84

4.7.4	Image RAM Addresser, EPM5128	84
4.7.4.1	Logic Overview	85
4.7.4.2	Interface with the Host PC and Offset DAC Control	87
4.7.4.3	Operation of the Image RAM Addresser.....	87
5	<i>Graphical User-Interface and Control Software</i>	90
5.1	Classes for GUI and Control Software	92
5.1.1	Overview	93
5.1.2	Classes for Graphical Functions	93
5.1.3	Classes to Handle User Activities.....	95
5.1.4	PassiveObject Class	96
5.1.5	ActiveObject Class	98
5.1.5.1	PopupWin Class.....	98
5.1.5.2	MenuBar Class.....	99
5.1.5.3	CommBox Class.....	100
5.2	Bitmap Editor	100
5.2.1	Overview	100
5.2.2	Management of Bitmap Files.....	101
5.2.2.1	PFileWin Class.....	101
5.2.2.2	File Functions	101
5.2.3	Manipulation of Bitmaps	103
5.2.4	Miscellaneous Functions.....	104
5.3	Main Control Routine.....	104
5.3.1	Overview	104
5.3.2	Set-up of DACs and Other System Parameters	105
5.3.3	Other Member Functions	107
6	<i>System Development, Integration, and Testing</i>	109
6.1	CCD Board	110
6.2	Modification of the System Hardware.....	112
6.2.1	Reduction of the Switching Noise	112
6.2.2	CCD Clock Drivers.....	113
6.3	Integration of System Modules	115
6.4	Performance of the Control System.....	117

6.4.1	Functional Verification	117
6.4.2	Image Acquisition Test	118
6.4.3	Linearity and Mean-Variance Tests	120
6.5	Suggestion for Future Work	123
7	Conclusion	130
	Bibliography	134
	Glossary	144
	Acknowledgements	146
	Appendices	
A	Logic Implementation of the Sequencer and the Image RAM Addresser	
B	Class Header Files for the GUI and Control Software	
C	Characterisation of a CCD Imaging System	

List of Tables

1.1	Adaptive optics systems for ground-based telescopes in various stages of development.....	21
2.1	Comparison of the CCD with two traditional means of imaging	25
2.2	List of some high-speed CCDs commercially available	33
3.1	Atmospheric seeing parameters at different wavelengths.....	35
3.2	Design goals of the ACCD.....	36
3.3	List of some commercially available CCD/MOS cameras with high frame rates in ascending order of data rate	53
6.1	Pin configuration of the Loral CCD on the CCD Board.....	111
6.2	Recommended operating condition of most commercial full frame CCDs from Loral Fairchild.....	119
6.3	Test operating condition of the Loral CCD from the Institute of Astronomy in the University of Cambridge.....	119

List of Figures

1.1	Block diagram of the adaptive optics system of a ground-based astronomical telescope	12
1.2	Principle of three types of wavefront sensors used in adaptive optics systems	20
3.1	A typical stream of Bitmaps to drive four CCD clock gates.....	43
3.2	Relationship between the binary counter and the external memory device in the Bitmap control method.....	44
3.3	Provision of clock pulses with the static DAC and Bitmap control methods ...	47
3.4	Required frame rate for the high-speed videography	49
4.1	Sequential generation of the Bitmaps to acquire an image frame from a frame-transfer CCD with the purging disabled	55
4.2	Block diagram of the prototype hardware of the high-speed CCD control system	56
4.3	Format and pin definition of the Loral 64×64 frame-transfer CCD.....	57
4.4	Layout of the Bitmap Generation Board	61
4.5	Organisation of each Bitmap RAM and its mapping by the sequencer	70
4.6	CCD readout procedure with the purging disabled.....	72
4.7	Layout of the Bias Board	75
4.8	Diagram of the translation and amplification circuit to supply the CCD bias from the DAC output.....	76
4.9	Layout of the Clock Board assigned to the vertical transfer clock gates	78
4.10	A typical horizontal transfer clock pulse stream for a 3-phase CCD.....	79
4.11	Circuit diagram of the vertical CCD clock driver.....	81
4.12	Circuit diagram to process and sample the CCD output video signal	83
4.13	Layout of the Signal Processing Board #1	86

4.14	Layout of the Signal Processing Board #2	89
5.1	Components of the GUI and control software and their interaction with the user and the system hardware	91
5.2	Hierarchical tree and relationship of classes for the GUI and control software.....	94
5.3	A portion of the horizontal transfer Bitmap file	102
5.4	Bitmap Editor when the Open item in the File menu is selected	103
5.5	Modification of the output voltage level of a LOW clock DAC.....	106
6.1	Circuit diagram of the AC-coupled pre-amplifier.....	112
6.2	Circuit diagram of the former clock driver built with the high- speed MOSFET driver EL7212	115
6.3	Photograph of the complete control system with the main power supply unit and the host PC.....	116
6.4	Response of the Loral CCD and the control system to different exposure levels.....	121
6.5	Response of the Loral CCD and the control system to different exposure time that was controlled by the frame-transfer Bitmap	126
6.6	Plot of the pixel variances against the mean pixel values for the data points in Figure [6.5].....	126
6.7	Two sequences of image frames acquired from the Loral CCD by the control system in the burst readout mode.....	127
6.8	Streams of clock pulses for the frame, vertical, and horizontal transfers used in the mean-variance test.....	128

Introduction

1.1 Adaptive Optics

The role of most astronomical telescopes is to collect as much light as possible from objects and to form as sharp images as possible. The light-collecting power of a telescope is a function of its aperture area and ideally, the sharpness of images or its angular resolution is a function of its aperture diameter limited by the diffraction. In practice, however, the air turbulence in various layers of the Earth's atmosphere causes wavefront distortion of the incoming light from an object and the blurring of its image. This phenomenon is referred to as atmospheric *seeing*.

The turbulent fluctuation of wind velocities in the upper atmosphere causes the fluctuation of the refractive index of the air. As the result, the incoming wavefront has spatial and temporal variations in phase and amplitude. The amplitude variation, known as scintillation, causes the intensity variation of the image whilst the phase variation causes its motion and broadening. The amplitude variation is generally ignored because the phase variation is dominant for the image quality. For a 4-meter ground-based telescope, the atmospheric seeing effect degrades its spatial resolution by more than one order of magnitude compared with its diffraction limit and the peak intensity of the image by a factor of about 100. The Hubble Space Telescope(HST) has demonstrated the achievable image quality in the absence of the atmospheric seeing effect.

Adaptive optics can be defined as the technique that corrects primarily the wavefront distortion introduced by the Earth's atmosphere and thereby attempts to restore the

image, in real-time, to a quality close to the diffraction limit of a telescope[47]. An adaptive optics system consists of three main elements; wavefront sensor(WFS), wavefront correction device, and servo system as depicted in Figure [1.1]. The WFS measures the distortion of the incoming wavefront and the servo system drives the wavefront correction device from the measured wavefront. The combination of a flat steering mirror (tip-tilt mirror) and a deformable mirror located behind the focus of the telescope is commonly used as the wavefront correction device that re-shapes the distorted wavefront. The deformable mirror is either continuous or segmented and is normally driven by a number of actuators. For the detection of the wavefront distortion, natural or artificial (laser) reference (or guide) stars in the isoplanatic angle (or patch) around the astronomical object are used. The isoplanatic angle is simply defined as the distance on the sky over which the wavefront distortion is practically constant.

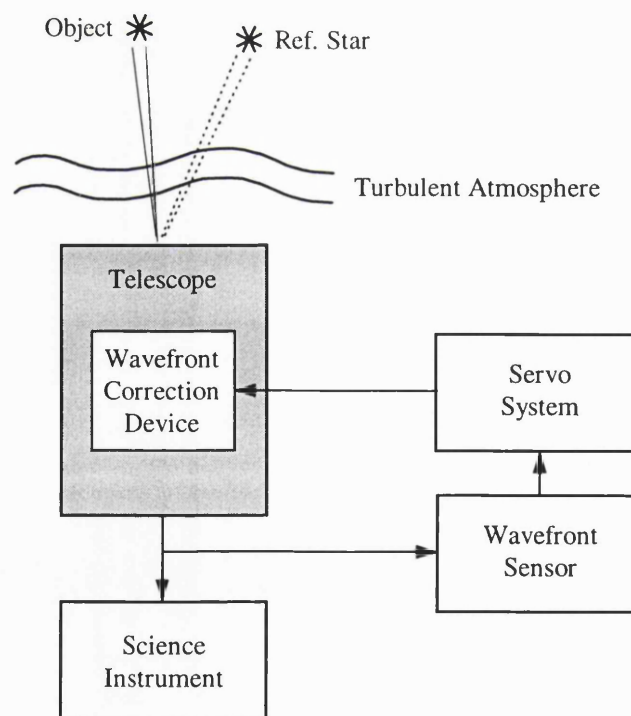


Figure [1.1] Block diagram of the adaptive optics system of a ground-based astronomical telescope

Being proposed by Babcock in 1953 for the first time[5], adaptive optics was applied in the US military to monitoring satellites and focusing lasers onto distant targets. The attempt to apply the adaptive optics to astronomical telescopes was very limited up to the mid 80's. Currently, there are numerous adaptive optics systems for astronomical telescopes in various stages of development around the world, some of which are listed in Table [1.1]. Beckers[7] attributes the recent surge of interest in the adaptive optics for astronomical telescopes to the following factors.

- *The attainment of the quantum limits of astronomical panoramic detectors which led to the construction of large telescopes with good image quality*
- *The realization that many of the complexities and limitations of the adaptive optics disappear at infrared(IR) wavelengths*
- *The recent availability of IR panoramic detectors*
- *The invention of the laser guide star for wavefront sensing*

The atmospheric seeing effect is less significant in near-IR(NIR) than in visible bands and it virtually disappears in mid IR bands. In NIR, the sizes of the isoplanatic patch and the subaperture are larger (lower spatial resolution) and the variation of the atmospheric seeing effect is slower (lower temporal frequency). Therefore, fainter reference stars can be used for wavefront sensing. In addition, the adaptive optics system becomes less complex and costly. The atmospheric seeing effect over different wavelengths can be examined in Table [3.1]. It shows that the isoplanatic angle over which reference stars are selected also increases with wavelength. Therefore, the statistical probability to find a reference star that is sufficiently bright increases with wavelength. The limitation imposed by the isoplanatic angle can be overcome by the use of artificial reference stars generated by lasers, which are commonly referred to as laser guide stars.

Beckers[7] also predicts that the adaptive optics system will be valuable in many fields of astronomy such as direct imaging, resolving complex configurations, detection of faint point sources, observation of faint companions, coronagraphy, spectroscopy of point-like objects, and interferometric imaging. For example, the COME-ON system on the 3.6-m ESO(European Southern Observatory) telescope produced the diffraction-limited images of various objects such as the asteroid Ceres

and the star η Carinae[7]. The ChAOS system on the 3.5-m Apache Point telescope also produced resolved images of the binary star δ Equ. In addition, the FASTTRAC II system with a sodium laser beacon obtained corrected images of the core of M13 (a globular cluster) on the MMT(Multiple Mirror Telescope). The number of astronomical papers related to observations made with adaptive optics also increased from 3 in 1993 to 28 in 1997¹.

1.2 Wavefront Sensor

Three different types of WFS's are widely used for the adaptive optics telescope. They are the lateral shearing interferometer, Shack-Hartmann sensor, and curvature sensor. Figure [1.2] is reproduced from Rousset[99] to illustrate their principles. In the lateral shearing interferometer, the incident wavefront is shifted (or sheared) and mixed with the original wavefront to form interference patterns at the overlap area. The interference patterns give the first derivative of the wavefront over the shear distance in the shear direction. To measure the wavefront completely, the wavefront is split into two channels, one of which is fed to the x -shear device and the other to the y -shear device.

In the Shack-Hartmann sensor, an array of lenslets is placed at an image of the telescope pupil. When the wavefront is distorted, each lenslet forms an image out of its optical axis on its focal plane. The position of the image gives the vectorial wavefront tilt in the subaperture covered by each lenslet as illustrated in Figure [1.2.b]. Therefore, the x and y slopes of the wavefront are simultaneously determined. Each subaperture needs at least four detector cells to determine the position of the image. The Shack-Hartmann sensor is the most commonly used WFS.

The curvature sensor consists of two image detectors; one records the illumination intensity at a distance before the focal plane and the other behind the focal plane at the same distance. Figure [1.2.c] illustrates that a local wavefront produces an excess of illumination at one plane and a lack of illumination at the other. By approximation, the difference between the illumination intensities on two detectors is proportional to

¹ Refer to <http://www2.keck.hawaii.edu:3636/realpublic/ao/>

the local wavefront Laplacian inside the light beam and to the wavefront radial first derivative at the edge of the beam. The wavefront can be reconstructed by solving the differential equation. This sensor can be directly coupled to a membrane or a bimorph mirror for wavefront correction. For the description and comparison of different WFS's in more detail, refer to Rousset[99] and Pennington *et al.*[88].

The WFS needs to sample the wavefront with enough spatial resolution and sampling frequency for the real-time correction of the distorted wavefront. The required spatial resolution and sampling frequency is a function of the atmospheric seeing conditions at a specific observing site and the scientific goal of the telescope. Chapter 3 briefly describes the required spatial resolution and sampling frequency. The spatial resolution and sampling frequency of various present or planned adaptive optics systems can be deduced from their numbers of mirror actuators and detector frame rates given in Table [1.1].

The WFS must be highly sensitive to work on faint objects with many subapertures. Therefore, detectors with high quantum efficiency(QE) or detectivity are employed. To achieve the high signal-to-noise ratio(SNR) at high frame rate, detectors with multiple output nodes working in parallel are frequently used. The WFS also needs to operate broadband with white light and on extended objects. Generally, the adaptive optics system measures the wavefront distortion at visible wavelengths for its correction in NIR or IR. This concept of polychromatic wavefront sensing is feasible because the refractive index of the air shows little variation between visible and IR bands.

As Table [1.1] shows, the charge-coupled device(CCD) is the most widely used detector for WFS's, especially for Shack-Hartmann sensors. Because the curvature sensor needs less detector cells per subaperture, avalanche photodiodes(APD) of high QE are frequently employed. They are shot-noise-limited detectors. The CCD has the high QE over wide bands of visible and NIR and a 100 % fill-factor. Its pixel positions are also precisely defined. Either intensified or bare, the two-dimensional CCD is ideal to measure the position of the image formed by each lenslet in the Shack-Hartmann sensor. Commonly, frame-transfer CCDs of relatively small formats are used to meet the required spatial resolution. They have a minimum time gap

between the wavefront measurement and correction. In 1990, for example, Clampin *et al.*[23] used a quadrant CCD developed by Tektronix and in 1994, Levine *et al.*[64] developed the ACCD(Adaptive Optics CCD) for wavefront sensing. Table [1.1] lists two more CCDs manufactured by Lincoln Lab. and EEV. The CCD used in the work of this thesis is also a frame-transfer CCD developed exclusively for wavefront sensing by Loral.

1.3 High-Speed CCD Control System for Adaptive Optics

Over the past several years, substantial progress has been made at the Optical Science Laboratory(OSL), Dept. of Physics & Astronomy, University College London(UCL) in adaptive optics and in work on CCD control systems. A new type of WFS based on a pair of crossed bi-prisms was designed[91]. In this sensor, the crossed bi-prisms split the image horizontally and vertically to form quadrant pupil images. If a ray is tilted from its path, it will lead to an imbalance in the light intensity between pupil images. The imbalance is directly related to the local tilt of the wavefront. This sensor exhibits a number of advantages over the conventional Shack-Hartmann sensor; easy real-time modification in software of the number and arrangement of subapertures, no light loss as in the area between lenslets, and no mechanical adjustment problem. It demonstrated its usefulness as a tilt sensor combined with the ISIS(Intermediate-dispersion Spectrograph and Imaging System) active collimator mirror developed at OSL.

In addition, research on the adaptive secondary mirror is in progress. Conventional adaptive optics systems use relay optics to form an image of the telescope pupil on the wavefront correction devices mounted on the optical bench. Therefore, additional replay optics are required for scientific instruments and the WFS. In comparison, an adaptive secondary system promises greater efficiency and throughput, reduced emissivity, and reduced polarisation effect because it does not require any extra relay or re-imaging optics. These are critical parameters to astronomical telescopes in NIR or IR. An 1-m adaptive secondary mirror made of a thin faceplate and 90 magnetostrictive actuators was designed after extensive studies[9, 10, 11]. Currently, a 270-mm concave spherical mirror with seven actuators is under construction for

demonstration. It is planned to test the demonstration mirror on the 4.2-m WHT(William Herschel Telescope).

The successful demonstration of the prototype low-noise mosaic CCD control system verified the versatility and effectiveness of the new CCD control method elaborated at OSL[41]. The new control method defines waveforms to drive CCD clock gates and to perform other functions with streams of digits called *Bitmaps*. In this previous work, it was employed to supply streams of multiplexed clock and signal timing pulses to a mosaic of CCDs.

The WFS described above uses a conventional CCD camera and a frame grabber to detect quadrant pupil images. Consequently, the data acquisition speed is limited by the speed of the camera electronics and frame grabber. This limitation presented the necessity to develop a dedicated CCD control system that may satisfy the required sampling frequency and spatial resolution for wavefront sensing. In addition, it was perceived that it presented a good opportunity to demonstrate the usefulness of the Bitmap control method in high-speed control systems. The design goals of this work, the development of high-speed CCD control system for adaptive optics, are summarised below.

- Readout frequency of approximately 1000 Hz for a CCD of 100×100 pixels
- Versatile control system
- Modular and compact control system
- User-friendly system interface

The high-speed CCD control system will demonstrate the feasibility of the complete WFS and lead to the construction of a working system for the adaptive secondary.

It is also believed that the control system will be useful in other applications such as the quantised knife-edge testing of large optics at OSL. In this test method, a conventional CCD camera is used to capture the image of the test optics for each discrete movement of the knife[59]. The critical factor is the speed of the acquisition and analysis of the image data. It is hampered by two factors; firstly, the discrete vertical movement speed of the knife driven by a DC motor with an encoder and secondly, the amount of memory to store images. It is planned to improve the system

performance by using a new personal computer(PC) with a large amount of memory and a new analysis algorithm. It is also planned to attempt the continuous movement of the knife. If the new algorithm reduces the accumulation of numerical errors, a higher spatial profile of the optics can be achieved. Therefore, the test system will require a CCD camera that produces images with higher spatial resolution at higher frame rate than the current camera. There are numerous other scientific and engineering applications that require high-speed CCD control systems as described in Chapter 3.

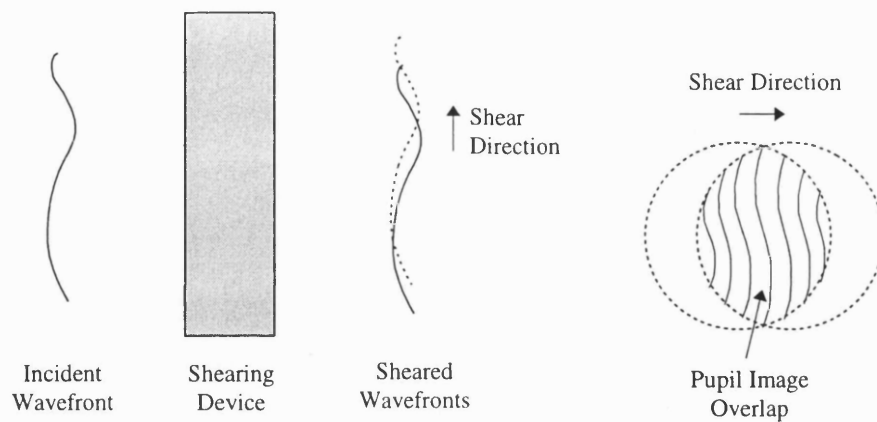
1.4 Contents of the Thesis and Author's Contribution

This thesis deals with the development of a high-speed CCD control system for adaptive optics. Much of the work described in this thesis was financially supported by the Satellite Technology Research Center(SaTReC) in the Korea Advanced Institute of Science and Technology(KAIST). Initially, it was planned to complete the testing of the control system and analysis of its results. However, before its completion, the author was summoned into the national service.

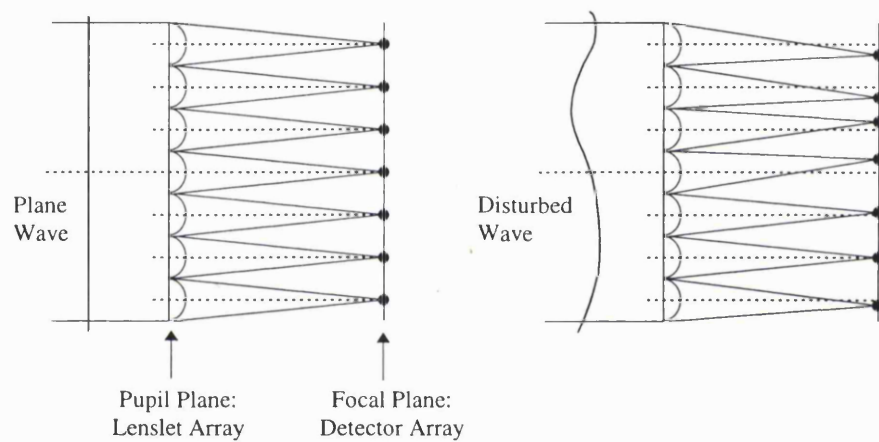
The contents of the thesis are as follows. Chapter 2 presents a brief history of the CCD, its physical characteristics, and a survey of some commercially available high-speed CCDs. In Chapter 3, following a brief review on some WFS's, the design goals of the high-speed CCD control system and the implementations made to meet them are discussed in more detail. In addition, a survey of some of high-speed solid-state cameras is presented and high-speed CCD systems in other applications are examined, such as motion analysis and high-speed image acquisition, high-speed photometry, and soft x-ray imaging. Chapter 4 gives the description of the prototype hardware of the control system following that of the Loral 64×64 frame-transfer CCD. Chapter 5 describes the graphical user-interface(GUI) and control software and various class objects developed for it. Chapter 6 deals with the integration of the high-speed CCD control system and its testing results with the discussion on them. Finally, in Chapter 7, the results of this work are summarised.

The author's contribution to the adaptive optics programme has been the development of the high-speed CCD control system, as follows:

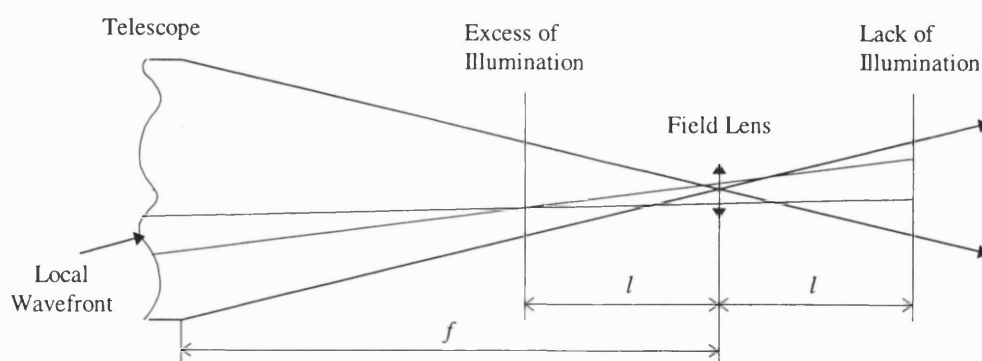
- Design and construction of all prototype hardware modules of the control system including the sequencer and the Image RAM Addresser
- Design and source-coding of all class objects and software routines for the GUI and the control software except for the super-VGA graphics library
- Integration and testing of prototype hardware modules, the GUI, and the control software



a. Lateral shearing interferometer



b. Shack-Hartmann sensor



c. Curvature sensor

Figure [1.2] Principle of three types of wavefront sensors used in adaptive optics systems reproduced from Rousset[99]

Programme	Aperture	Observation band	Mirror type	No. mirror actuators	Wavefront sensor	References
ESO ADONIS ²	3.6 m	NIR	Tip-tilt Deformable	52	2 S-H Reticon for high flux, 200 Hz EBCCD ³ for low flux, 100 Hz	http://www.eso.org/aot/adonis/
W. M. Keck Observatory Keck II	10 m	NIR	Tip-tilt Continuous	349 (241 active)	S-H, LL ⁴ 64×64 CCD 11 e ⁻ noise at 2 KHz	[35] http://www2.keck.hawaii.edu/
Lick Observatory Shane Telescope by LLNL ⁵	3 m	NIR	Tip-tilt Continuous	127	S-H, LL 64×64 CCD	[83] http://ep.llnl.gov/
Steward Observatory FASTTRAC II (MMT)	6 × 1.8 m	NIR	Beam-combining pyramid (tip-tilt)	NA	S-H, 64×64 CCD	[37] http://athene.as.arizona.edu:8000/
Steward Observatory MMT Conversion	6.5 m	NIR	Adaptive secondary	324	S-H, EEV 80×80 CCD 1 KHz	[67, 100]
U. Chicago ChAOS ⁶	3.5 m	NIR	Continuous	201	S-H, LL 64×64 CCD 13 e ⁻ noise at 2 KHz	[105] http://astro.uchicago.edu/chaos/
Anglo-Australian Telescope	4 m	NIR	Tip-tilt Continuous	13	CS, APD	[85]

Table [1.1] Adaptive optics systems for ground-based telescopes in various stages of development (continued over)

Programme	Aperture	Observation band	Mirror type	No. mirror actuators	Wavefront sensor	References
CFHT AOB ⁷	3.6 m	Visible NIR	Tip-tilt Continuous	19	CS, APD 20 e ⁻ noise	[93] http://www.cfht.hawaii.edu/
UK Adaptive Optics ELECTRA (WHT)	4.2 m	NIR	Tip-tilt Segmented	228	S-H, 2 CCDs 7 e ⁻ noise at 1 M pixels/sec/port	[34, 81] http://ukao.dur.ac.uk/
Mt. Wilson Observatory	100 in	Visible NIR	Tip-tilt Deformable	241	S-H, ADAPT2 64×64 CCD ⁸ 7.6 e ⁻ noise at 1600 Hz	[102, 110] http://www.mtwilson.edu/
Starfire Optical Range	1.5 m	NA	NA	NA	S-H, 2 LL 64×64 CCDs	[89] http://www.sor.plk.af.mil/

Table [1.1] Adaptive optics systems for ground-based telescopes in various stages of development

Visible : 400 - 900 nm, NIR : 1 - 5 μ m, S-H : Shack-Hartmann sensor, CS : curvature sensor

The survey was conducted in mid 1997. The 'wavefront sensor' column lists the type of WFS, the detector, the readout noise, and the readout frequency. The last column gives the WWW(World Wide Web) addresses in addition to some references.

² ADaptive Optics Near Infrared System, upgrade of COME-ON PLUS

³ Electron Bombarded CCD

⁴ Lincoln Lab., Massachusetts Institute of Technology

⁵ Lawrence Livermore National Laboratory

⁶ Chicago Adaptive Optics System

⁷ Canada-France-Hawaii Telescope Adaptive Optics Bonette

⁸ ACCD with floating diffusion output amplifiers

Charge-Coupled Device

The concept of the charge-coupled device(CCD) was devised and verified experimentally with a linear array of MOS(metal-oxide-semiconductor) capacitors for the first time by Boyle and Smith[13, 14]. In this experiment, they used MOS capacitors on an n-type silicon substrate and holes as minority carriers to transfer the information along potential wells. As for applications of this concept of a new semiconductor device, they clearly saw the potential of two-dimensional CCDs as imaging devices[13]:

'The basic shift register concept may be used to construct a recirculating memory or used as a delay line for times up to the storage time. Clearly, charge transfer in two dimensions is possible as well as the ability to perform logic. An imaging device may be made by having a light image incident on the substrate side of the device creating electron-hole pairs. The holes will diffuse to the electrode side where they can be stored in the potential wells created by the electrodes. After an appropriate integration time, the information may be read out via shift register action. A display device may be constructed by the inverse process of reading in the information (minority carriers) via shift register action and then forward biasing the MIS structure to force the minority carriers into the bulk where radiation recombination takes place.'

In the beginning, it was generally thought that the CCD technology would prevail in the development of high density memories. The photosensitivity of the CCD was

employed as a tool to evaluate the progress of technology development in the semiconductor manufacturing[118]. However, the technological advance to develop high-density CCD memories could not follow the pace of the explosive advance of NMOS dynamic RAMs technology. As the result, the emphasis of the CCD technology moved to imaging applications and the photosensitivity became the driving force of its continuous technological development. Not being the driving force in the semiconductor market, the fact that CCD imaging devices are more sensitive to semiconductor process variations contributed to the understanding of underlying physical phenomena and to the more rapid advance of the dynamic RAM technology.

In 1985, it was reported that about 20 companies were heavily investing in the development of solid-state imaging sensors with an estimated cost of 100 million US dollars per year[25]. Videography and machine vision were two main applications driving this heavy investment. The dominance of the CCD as a scientific imaging device in astronomy is evident in two facts; Even as long ago as in 1988, approximately 97% of the observing-time proposals for ground-based telescopes requested CCDs[116], and the Wide Field and Planetary Camera(WFPC) in the HST employs eight CCDs to cover the spectral band from 0.12 μm to 1.0 μm [111]. The characteristics of the CCD compared with other two traditional imaging means are listed in Table [2.1] reproduced from Gursky[40].

In section 2.1, some principal physical characteristics of the CCD are presented. The role of the CCD in x-ray and IR bands is also briefly described. Whilst x-ray CCDs and are not relevant to the adaptive optics application of this thesis, their description is included to highlight the usefulness of high-speed CCDs and thus the high-speed CCD control system. The general description of IR detectors is included to show the usefulness of the CCD not only as the detector but also as the multiplexer. In section 2.2, the characteristics of high-speed CCDs are given, which is followed by a survey of some commercial high-speed CCDs.

	CCD	Intensified Imaging	Photographic Emulsion
Quantum Efficiency	high (~50 %)	medium (~10 %)	low (~few %)
Stability	good	poor	poor
Gain	low (~1)	high ($\sim 10^6$)	high ($\sim 10^{12}$)
Number of pixels	low ($10^5 - 10^6$)	low ($\sim 10^6$)	high ($10^8 - 10^9$)
Commercial use	wide	limited	wide

Table [2.1] Comparison of the CCD with two traditional means of imaging reproduced from Gursky[40]

2.1 Characteristics of the CCD

2.1.1 Quantum Efficiency

The quantum efficiency(QE) of a detector is defined as the average number of free electrons produced per incident photon. Two physical parameters limit the QE of silicon CCDs. Theoretically, the silicon band-gap energy of 1.14 eV limits the IR sensitivity to about 10^4 Å and its density gives the high-energy cut-off wavelength of about 1 Å. CCDs made from other materials rather than silicon can have extended spectral responses. For example, germanium CCDs can cover the spectral band from 0.5 to 1.7×10^4 Å, approximately.

The QE of frontside-illuminated CCDs within 10 and 6000 Å is largely dependent upon the transparency, reflectance, and surface conditions of layers overlying the photosensitive volume on the bulk substrate[52]. Out of this boundary, the QE depends on the thickness of the photosensitive volume. To obtain high QE and to exploit the advantages of the multi-gate technology, the CCD substrate can be thinned to allow the illumination from its backside. The backside surface must be treated following the thinning process to avoid two undesirable properties; the positive charging of the backside surface and QE hysteresis(QEH)[53]. They are caused by the loss of signal electrons through recombination in the native oxide layer formed at the backside. QEH is the phenomenon that the QE increases when signal electrons are

trapped at the backside surface and decreases when the trapped electrons recombine over time.

Currently, four backside treatment techniques are used to charge the backside surface negatively in order to repel signal electrons; backside charging, flash gate, biased flash gate, and boron coating. The backside charging technique involves the negative charging of the surface with an intense ultra-violet(UV) light flood. Being effective, it requires periodic re-charging because backside charges dissipate when the CCD is cooled. In the flash gate technique, a thin film of a high work-function metal is deposited on the backside. It supplies electrons to the backside surface and can also be used for the anti-reflection coating to increase the QE. Under high vacuum conditions, however, additional positive charges built up at the backside surface neutralise electrons supplied by the flash gate. The flash gate is biased slightly negative to compensate for this build-up in the biased flash gate technique. Boron implants coated at the backside provide a permanent hole layer, which generates similar conditions as in other treatment techniques. Refer to Janesick *et al.*[54] for the description of the backside treatment in detail.

However, backside thinning and treatment involves complex manufacturing processes and high cost. Alternatively, the virtual phase CCD provides improved QE at shorter wavelengths by minimising the absorption of photons by the frontside layers with implanted gate technology. It only requires a single clock gate to perform the charge transfer. A simpler method to improve the QE of frontside-illuminated CCDs at shorter wavelengths is to use *down-converter* materials such as fluorescent dyes and phosphor. Coated on the frontside, they convert incident UV photons into photons of longer wavelengths.

Greater improvement of the QE can be achieved with the deep depletion at both ends of the spectral response range of silicon CCDs. Deep depletion CCDs are fabricated on high-resistivity silicon that deepens potential wells. They also prevent the diffusion of signal electrons in an uncontrolled direction and lead to higher charge collection efficiency.

2.1.2 Readout Noise

The dominant noise in CCDs at high illumination levels should be photon shot noise (assuming the pixel-to-pixel variation is removed). At low illumination levels, the CCD is dominated by intrinsic device noise sources such as trapping-state noise, reset noise, background noise, charge transfer noise, and output amplifier noise. The trapping-state noise is caused by the trapping and slow release of the signal charges in potential wells either by surface or bulk trapping states. The buried-channel operation prevents the interaction of signal charges with surface trapping states whereas the density of bulk trapping states can be decreased by a good materials control in the fabrication process.

Background charges can be added to the signal charge from dark current, residual image, luminescence, and fat zero. The dark current is due to electrons thermally generated within the CCD and it represents an inherent noise floor. They can be generated in bulk and depletion regions and at the Si-SiO₂ interface. If the CCD is operated in the inverted mode where the surface potential at the Si-SiO₂ interface is lower than the substrate potential, the dark current generation at the interface can be reduced because holes migrated from channel-stops fill the interface states. The operation in total inversion also eliminates the residual image that occurs when the CCD is over-exposed or when the buried n-channel becomes undepleted. CCDs that can operate with all phases of clock gates totally inverted are referred to as multi-pinned phase(MPP) CCDs. They require the silicon to be doped with boron to achieve the well capacity in total inversion. However, total inversion results in lower full-well capacity than the normal mode or the partial inversion.

There are four luminescence sources; clocking, diffusion, pinch-off, and blemish luminescence. Clocking the horizontal register over a long period of integration causes its nearby pixels to acquire some additional charges : clock luminescence. The diffusion luminescence is related to the n-p diffusions in the input-output structure of the CCD whereas the pinch-off luminescence is associated with the channel of the output MOSFET(MOS field-effect transistor). The blemish luminescence is related to a single pixel resulting in the saturation of the CCD in a few seconds. The fat zero is often introduced either optically or electrically to fill in the trapping states and to

increase the charge transfer efficiency(CTE) of a CCD. However, it inevitably contributes to the increase of the noise.

Together with the noise due to the finite CTE, the spurious charge noise is a noise related to the charge transfer mechanism. It is caused by holes migrated from channel-stops and trapped at the Si-SiO₂ interface when a CCD is driven into total inversion. It can be reduced by shaping CCD clock pulses to have longer rise times and lower voltage swings or by employing the tri-level clocking technique where clock pulses have an intermediate level between their HIGH and LOW voltage levels.

The ultimate readout noise floor of a CCD is limited by noise in the on-chip amplifier; thermal white and flicker (1/f) noises. They are primarily dependent upon the size of the amplifier and can be reduced by constructing a physically larger amplifier. However, it decreases the output sensitivity because the input capacitance of the amplifier increases as it becomes larger. Janesick and Elliott[52] stated that the sophisticated optimisation of parameters associated with the on-chip amplifier might reduce the readout noise of a CCD down to the level of 1 electron rms at slow scan operation. In addition, a non-destructive readout technique that enables the multiple detection of signal charges can lower it effectively below the level of 1 electron rms[122].

2.1.3 Charge Transfer Efficiency

The charge transfer efficiency(CTE), a measure of the ability to transfer the signal charges from one potential well to the next, is mainly influenced by spurious potential pockets. They cause the loss of signal charges by the improper well shape and depth beneath pixels. They can be introduced in the design and fabrication process of a CCD apart from those caused by impurities and defects in the silicon. Their influence can be reduced by maximising the transfer time, increasing clock voltage levels, and operating the CCD at higher temperatures. Ideally, however, they must be eliminated in the design and fabrication process. Energetic particles and photons also generate spurious pockets by disrupting the lattice structure of the silicon. With the current technology, the CTE for scientific CCDs can reach 0.9999995, which means that about 99.5% of the signal charges remains after the charge transfer along 10^4 pixels.

2.1.4 Physical Size

The production yield during manufacturing in terms of the number of gate shorts and opens directly limits the physical size of CCDs. It was reported that the device yield had improved to build wafer-scale CCDs[51]. For example, a 7000×9000 CCD with $12 \mu\text{m}$ pixels that occupies an entire 6-inch silicon wafer was in the planning stage.

An alternative approach to construct a larger photosensitive area with CCDs of smaller sizes is to mosaic them[41]. For example, the design of an 8192×8192 mosaic CCD with 3-edge-butable 2048×4096 CCDs was described by Luppino *et al.*[71] They also listed numerous examples of mosaic CCDs built or planned for astronomical observations. This approach has the advantages that different CCDs can be mosaicked and damaged CCDs can be replaced readily. However, the packaging cost from the tight tolerance requirement in butting them must be taken into consideration. It may be more cost-effective to fabricate a full wafer CCD than to mosaic several CCDs.

2.1.5 CCDs in X-ray

X-ray detectors must be able to handle large photon fluxes, have fast data handling capability, and give high spatial resolution. The most widely used detectors in soft and intermediate x-ray energy bands are CCDs, microchannel plates, and proportional counters. Microchannel plates give excellent spatial resolution, but have poor efficiency (~ 20 %) and fail to produce any information on the energy resolution. Proportional counters have excellent efficiency, give moderate spatial and energy resolution, but do not have the intrinsic background-rejection capability. In comparison, CCDs provide excellent position resolution and good energy resolution and have good efficiency and the intrinsic background-rejection capability.

Many x-ray CCD detectors employed phosphor or fluorescent dyes to convert x-rays to visible photons. Fibre-optic tapers were also frequently used as face plates. However, the use of these down-converters can be detrimental for scientific applications that require the fast readout and moderately high spatial resolution. CCDs

for visible applications normally have the QE of 10 to 30 % for photons between 1 keV and a few keV. The direct detection of soft x-rays with CCDs was made feasible with the development of a new type of CCD that has longer absorption depths, i.e., deep-depletion CCDs. With the current manufacturing technology, it is possible to produce depletion layers as thick as 50 μm using very low dopant concentrations (the high-resistivity silicon). These deep-depletion CCDs have the soft x-ray QE reaching 100 %, although at the expense of decreased spatial resolution.

One of the demanding challenges for CCD x-ray detectors is the acquisition and analysis of the image data at high speed. In structural biology and protein crystallography, for instance, target samples deteriorate rapidly under high radiation exposures. Therefore, it is essential to read out the CCD at high speed. In protein crystallography and medical imaging, increasing the physical size of CCDs is another important requirement. An array (or a mosaic) of CCDs coupled with fibre-optic tapers can satisfy such a requirement although, the physical size is limited by the largest fibre-optic taper available in this approach.

2.1.6 CCDs in Infrared

CCDs in the visible band have been developed using silicon which has the band-gap energy of 1.14 eV both as the detector and the multiplexer material. IR detectors with high sensitivity, operating in atmospheric transmission bands of 3 to 5 and 8 to 12 μm , require other detector materials with narrower band-gap energies of 0.25 to 0.1 eV. Because the IR image of a typical terrestrial scene gives a very high background floor, an IR CCD needs to create deep potential wells with high potentials. High potentials in narrow band-gap materials cause a significant increase in the number of electrons excited into the conduction band. As the result, the noise increases and the storage capacity decreases. Therefore, IR detectors have been typically fabricated with narrow band-gap detector materials and silicon multiplexers such as CCDs.

Monolithic IR detectors perform the photon detection and the signal multiplexing (or readout) within the same material. They allow fabrication using silicon-like wafer processing and devices of lower cost. Fully monolithic IR detectors that are compatible with visible CCDs in design have been fabricated with PtSi or extrinsic Si.

Some effort has been focused on the development of fully monolithic detectors in the CCD structure with materials such as HgCdTe or InSb. Many other materials require constant bias operation that is incompatible with the CCD structure or require bias currents too large to handle. Another approach is the hetero-epitaxial growth of a narrow band-gap material on a silicon substrate that contains the multiplexer. It combines the best features of the mature silicon technology with narrow band-gap materials in a fully monolithic design.

Hybrid IR detectors perform the photon detection in a narrow band-gap material and the transfer of signal charges in the silicon. They have the distinct advantage of storing charges and performing multiplexing in the conventional silicon devices. To interface two different materials, the indium bump and the via-hole techniques are used. The indium bump technique involves mating each detector cell to the silicon multiplexer with previously deposited indium bumps fabricated on both detector array and silicon multiplexer. However, it may be difficult to reduce the size of the unit cell below 25 μm . The via-hole technique involves highly polishing both detector array and silicon multiplexer surfaces and gluing them together. The single chip produced in this way can be processed in a silicon-like fashion. However, via-holes apparently reduce the fill-factor.

Other readout structures competing with the CCD for IR imaging are the MOSFET switch, charge-injection device(CID), and charge-injection matrix(CIM)[101]. In the MOSFET switch structure, each pixel can be addressed individually. A column of detector diodes and storage capacitors is selected sequentially by the scan register and a row is selected by the vertical scan register. In the CID, the photon detection occurs within unit cells comprised of two MIS(metal-insulator-semiconductor) structures that are read out in an x-y addressable manner. This structure is similar to the MOSFET switch, but has extra switches on row and column lines to control the charge transfer within unit cells. The CIM structure consists of an array of MIS detectors wire-bonded to silicon processing chips in a pseudo-monolithic way. Narrow diode channels are used as output nodes to drain charges collected in individual MIS detector cells. The readout is generally performed by electrically scanning the MIS detector rows while synchronously sensing and multiplexing output voltages of the diode columns.

2.2 CCDs for High-Speed Applications

The speed of vertical transfer of the signal charges in a CCD is usually limited by the RC time constant associated with clock gates. The on-chip power dissipation can also be a limiting factor, especially when the high-speed vertical transfer is continued for a large fraction of the total readout time. The speed of charge transfer in the horizontal register is fundamentally limited by the velocity of electrons in the silicon. However, there is a practical limit set by the difficulty to generate and transmit clock pulses in high speed. For a given size of pixel, the operation at higher speed can be achieved using a deeper buried channel or higher clock voltage levels. It must be noted, however, that deeper buried channels require special processing and increased clock voltage levels result in higher power dissipation.

For high-speed CCDs, one or more additional MOS source followers are usually incorporated to interface with the external circuitry (pre-amplifiers). They provide the maximum bandwidth for a given transistor type and do not require the complexity of the DC-bias stabilisation circuit. Another important consideration on the output amplifier is the output slew rate. The output quiescent current must be sufficient to supply the charging current to the load.

Table [2.2] lists some of commercially available CCDs that are designated for high-speed applications by manufacturers. The principal characteristic of many high-speed CCDs is that they have multiple output nodes associated with sub-sections of their pixels. Such CCDs are generally referred to as multiport CCDs. It contributes to a significant reduction of the total readout time. Although incorporating more output nodes can lower the clocking frequency to achieve a specific readout time, the additional power dissipation and extra circuit complexity also needs to be considered. Like other scientific CCDs mentioned in section 1.2, EEV CCD39 is a high-speed CCD developed specifically for tracking and adaptive optics applications.

Mfg.	Model	Format	Frame Rate (fps)	Operation Mode	No. Output Nodes
Dalsa	IA-D1 Series	32×32, 64×64, 128×128, 256×256	10000 for 32×32 CCD	Frame Transfer	1
	IA-D2-0512	512×512	53	Full-Frame	1
	IA-D6	512×512	200	Frame Transfer	4
EEV	CCD39	80×80	1000	Frame Transfer	4
	CCD13	512×512	500	Full-Frame	8
	CCD17	512×512	200	Frame Transfer	2
	CCD37	512×512	200	Frame Transfer	2
EG&G Reticon	HS0512J	512×512	1000	Frame Transfer	16
	HS0256J	256×256	1000	Frame Transfer	4
Thomson-CSF	TH7895A-(H)	512×512	100	Full-Frame	2
	TH7896A-(H)	1024×1024	36	Full-Frame	4
	TH7887A	1024×1024	60	Frame Transfer	4
Texas Instruments	TC213	1024×512	30	Frame Transfer	2

Table [2.2] List of some high-speed CCDs commercially available[1, 20, 21, 49]
The survey was conducted at the beginning of 1996. The information on EEV CCDs is from data sheets that are not given in Bibliography and the information on TC213 is from <http://www.ti.com/sc/>.

High-Speed CCD Control System for Adaptive Optics

As pointed out in Chapter 1, the required spatial resolution and sampling frequency for a WFS is a function of the atmospheric seeing conditions and the scientific goal of the telescope. The atmospheric seeing effect over different wavelengths can be observed from the various parameters listed in Table [3.1] that is reproduced from Beckers[7]. r_0 is the coherent length or the Fried parameter that is defined as the characteristic size over which the mean square phase variations of the wavefront in a circular aperture are just over 1 radian². It can be thought as the maximum diameter of a telescope that produces diffraction-limited images. τ_0 is the typical parameter used to characterise the temporal variations of the wavefront and θ_0 is the isoplanatic angle as defined in Chapter 1. τ_{det} is the photon integration time of Shack-Hartmann sensors with CCDs. These parameters are given by

$$r_0 = 0.185 \lambda^{6/5} \cos^{3/5} z \left(\int C_n^2(h) dh \right)^{-3/5},$$

$$\tau_0 \approx 0.314 r_0 / V_{\text{wind}}, \quad f_0 = 1/\tau_0,$$

$$\theta_0 = 0.314 r_0 / H,$$

$$\tau_{\text{det}} \approx 0.3 \tau_0, \quad f_{\text{det}} = 1/\tau_{\text{det}}$$

where λ is the wavelength, z is the zenith angle, $C_n^2(h)$ is the refractive index structure constant at the altitude h , V_{wind} is the wind velocity, and H is the average altitude at which the atmospheric seeing occurs[7].

Spectral band	λ (μm)	r_0 (cm)	τ_0 (sec)	f_0 (Hz)	τ_{det} (sec)	f_{det} (Hz)	θ_0 (arcsec)
U	0.365	9.0	0.009	111.1	0.0028	357.1	1.2
V	0.55	14.9	0.015	66.7	0.0047	212.8	1.9
I	0.90	27.0	0.027	37.0	0.0085	117.7	3.5
H	1.62	55.0	0.055	18.2	0.0173	57.8	7.0
L	3.4	133.0	0.133	7.5	0.0418	23.9	17.0
M	5.0	210.0	0.21	4.8	0.0659	15.2	27.0

Table [3.1] Atmospheric seeing parameters at different wavelengths (0.75 arcsec seeing at 0.5 μm ; $V_{\text{wind}} = 10$ m/s; $H = 5000$ m)

The atmospheric seeing in arcsec is approximately given by λ/r_0 when the telescope diameter is larger than r_0 .

The Greenwood frequency is also often used to specify the 3-dB cut-off frequency of a simple closed-loop adaptive optics system. Greenwood showed the cut-off frequency may lie between 64 and 172 Hz in the near-worst-case atmospheric seeing conditions when the wavefront correction device is assumed to match the phase perfectly, at least spatially[38]. He also investigated two cases of observation; observation along a horizontal path and astronomical observation from a mountain-top. In both cases, he reported a cut-off frequency less than 200 Hz. The Greenwood frequency, f_G , is generally taken to be related to the Fried parameter by

$$f_G = 0.43 \frac{V_{\text{wind}}}{r_0} = 0.135 f_0.$$

For the Shack-Hartmann sensors, the dimensions of each lenslet at the telescope pupil plane are taken to agree with the Fried parameter at the given seeing conditions and wavelength. With the seeing conditions in Table [3.1] and a 10-m telescope diameter, for example, the required numbers of subapertures are approximately 4485 and 1366 at 0.55 and 0.90 μm , respectively. These figures are obtained by simply equating the entire telescope pupil area to the pupil area covered by a single subaperture multiplied by the number of subapertures. At least four pixels must be assigned to each subaperture in order to determine the vectorial position of the local wavefront.

Therefore, a 100×100 CCD gives enough spatial resolution to correct the distorted wavefront at $0.90 \mu\text{m}$ and longer. Because most adaptive optics systems listed are designed to work in NIR, the CCD control system with 100×100 pixels can be applied to them. Some WFS's assign 4×4 pixels to each subaperture in order to determine the position more precisely or to reduce the cross-talk between neighbouring pixels. In this case, a 100×100 CCD meets the required spatial resolution for $1.62 \mu\text{m}$ and longer.

The figure often quoted to specify the required readout frequency of a modern WFS is 1000 Hz. As can be seen in Table [1.1], present WFS's that use CCDs are designed to have a readout frequency of the order of a few kHz. The readout frequency of 1000 Hz satisfies the requirement at visible wavelengths given in Table [3.1] and exceeds the maximum predicted by Greenwood. For reference, some of the design goals adopted to the ACCD are reproduced from Levine *et al.*[64] in Table [3.2]. Even being ambitious, they indicate the required attributes of a CCD and its control system for wavefront sensing.

Attribute	Quantity
Format	32×64 frame-transfer
O/P nodes	32 Skipper floating gate amplifiers at the end of each signal column
Pixel size	$30 \mu\text{m}$ nominal
Readout noise	$1.5 e^-$ rms noise floor
Frame rate	1500 Hz
QE	80 % goal at $0.6 \sim 0.7 \mu\text{m}$ & 70 % goal at $0.8 \mu\text{m}$
Max. signal	4000 electrons minimum & 10000 electrons goal
Frame-transfer time	Transfer time/exposure time; 3 % max., 1 % goal
Dark current	1 electron/25 ms per pixel at room temperature

Table [3.2] Design goals of the ACCD reproduced from Levine *et al.*[64]

To repeat, the design goals of the high-speed CCD control system are a readout frequency of approximately 1000 Hz for a CCD of 100×100 pixels, the construction

of a versatile, modular, and compact system, and the construction of a user-friendly interface. The previous description demonstrates that such a CCD control system is instrumental in wavefront sensing in NIR and IR. It will become evident later that its versatility also makes it useful in visible bands. Another important characteristic used to specify a WFS is its output noise level because natural reference stars are normally faint and the incoming light beam has to be divided between many subapertures. This issue is addressed in section 6.5.

It would be ideal to review other CCD control systems for WFS's and present an in-depth comparison between them and the control system described in this thesis. However, hardly any relevant material on such CCD control systems was available. Instead, in section 3.1, a brief review is made of some WFS's built or planned for astronomical telescopes. In section 3.2, the approaches elaborated in the work of this thesis to meet the design goals are examined and some of them are compared to other approaches because all CCD control systems have some common characteristics. The detailed description of the control system is presented in the following chapters; the system hardware in Chapter 4 and the control software and user-interface in Chapter 5, respectively. High-speed CCD systems used in some other scientific and engineering fields are also reviewed in section 3.3 because its versatility makes the control system described in this thesis applicable in these fields. They are the motion analysis and high-speed image acquisition, the high-speed photometry, and the soft x-ray imaging.

3.1 Wavefront Sensors for Astronomical Telescopes

There are various adaptive optics systems for astronomical telescopes in different stages of development or in the stage of conceptual design around the world. Some of them are summarised in Table [1.1]. The W. M. Keck Observatory was reported to be in the critical design phase of a natural and single laser reference star adaptive optics system for the Nasmyth platform of the 10-m Keck II telescope in collaboration with Lawrence Livermore National Laboratory(LLNL)[35]. Its WFS is based on a 64×64 Lincoln Lab.(LL) CCD that exhibits 11 electrons rms readout noise at 2 kHz frame rate. Another laser reference star adaptive optics system was developed by LLNL for

the 3-m Shane Telescope at the Lick Observatory[83]. The system uses a Shack-Hartmann WFS that is configured to have a triangular array of 37 subapertures across the telescope pupil. It also employs an imaging system based on an LL 64×64 CCD, which was manufactured by the Adaptive Optics Associates.

It was reported that the existing six mirrors of the MMT would be replaced by a single 6.5-m mirror in late 1997[67, 100]. In addition, an adaptive secondary mirror made of a 2-mm thick continuous facesheet would be installed. The adaptive optics system of the new telescope by the Steward Observatory would employ a Shack-Hartmann sensor with 150 subapertures and a EEV 80×80 CCD that has 4 output nodes. Its pixels would be binned by a factor of 2 and it would be operated at the pixel readout rate of 250 kHz for each output node. The readout noise of the WFS was estimated to have to be lower than 3 electrons rms. As a preliminary step for the replacement, the FASTTRAC II adaptive optics system was developed for the existing six mirrors[37]. Its WFS uses a 64×64 frame-transfer CCD mounted in a thermoelectrically-cooled dewar. The CCD pixels are binned by a factor of 4. Even if this system only performs the tip-tilt correction for primary mirrors, the WFS can be regarded to have six subapertures because the light beams from them are combined by the adaptive beam combining pyramid and fed to the WFS.

Other adaptive optics systems that employ Shack-Hartmann type sensors are the ADONIS of ESO, the ChAOS of the University of Chicago, the ELECTRA for WHT, and the ADOPTICS of the Mt. Wilson Observatory[34, 81, 102, 105, 110]. The ChAOS for the 3.5-m Apache Telescope uses a 64×64 LL CCD and its WFS features 256 subapertures, the frame rate of 2 kHz, and the readout noise of 13 electrons rms. The WFS of ADOPTICS employs the ACCD with 32 skipper output amplifiers and gives 7.6 electrons rms readout noise at the frame rate of 1.6 kHz. In case of the ELECTRA by the UK Adaptive Optics Programme, its WFS with two CCDs features 7 electrons rms readout noise at 1 MHz pixel sampling frequency for each output node. As a part of the site evaluation programme both for WHT and the UK Infrared Telescope(UKIRT) by the UK Adaptive Optics Programme, a 64-subaperture Shack-Hartmann sensor that has the frame rate of 850 Hz was developed and commissioned at WHT. It is an open-loop system that has the facility to store 27000 continuous image frames at burst readout mode. As for turn-key WFS systems, there are

WaveScope of Adaptive Optics Associations, Inc., CLAS-2D of Wavefront Sciences, and Detect 16 of Optical Test Instruments.

Two examples of adaptive optics systems that employ curvature WFS's are the AOB(Adaptive Optics Bonnete) for the Canada-France-Hawaii Telescope(CFHT) and the one planned for the Anglo-Australian Telescope(AAT)[85, 93]. The WFS of AOB has 19 subapertures and employs APDs that feature 45 % peak QE and the dark current of about 20 electrons/sec. The WFS envisaged for AAT consists of 13 APDs coupled to an array of lenslets by optical fibres. The Center for Astronomical Adaptive Optics(CAAO) at the Steward Observatory also developed a similar curvature WFS that uses 13 fiber-coupled APDs and a 13-electrode bimorph mirror.

3.2 Design Considerations

3.2.1 Readout Frequency

The performance of a CCD control system in terms of its readout speed is primarily limited by two factors; the speed of the *sequencer* and the switching frequency of the clock driver circuitry. The sequencer refers to the function module that is responsible for the generation of CCD clock timing pulses. In the beginning, the control system was designed based on the assumption that the CCD has the format of 100×100 pixels and a single output node (the worst case). In this case, the frame rate of 1000 Hz gives the pixel sampling frequency of approximately 10 MHz because the readout time associated with horizontal transfers of the signal charges is dominant in the total readout time. In order to provide clock pulses at this sampling frequency, the sequencer must operate at higher frequencies. Even if the sequencer has the capability to supply the CCD clock timing pulses at 10 MHz or higher, it must be accompanied by clock drivers that have the capability to convert them into actual clock pulses. It is critical to the overall system performance because CCD clock gates are intrinsically capacitive and they generally require large voltage swings.

Physically, a monolithic sequencer is built either with a processor(s) or with logic circuits. Sequencers based on micro-processors are found in [18], [95], and [29] and those based on DSPs(digital signal processors) in [58] and [61]. Sequencers based on

logic circuits can be implemented either with hard-wired circuits [12] or with FPGAs(field-programmable gate arrays) [26]. A unique sequencer is found in [125] where a programmable digital word generator is used to provide clock timing pulses.

Because of its inherent sequential nature, a sequencer is readily implemented with hard-wired logic circuits or an FPGA(s). Such sequencers exhibit the minimum overhead time, which is advantageous for the high-speed operation. However, they have less flexibility compared with those based on processors in terms of the ability to perform sophisticated functions. For example, their ability to perform the readout of a sub-section of a CCD is limited either by the complexity of the hard-wired logic circuits or by the logic density of the FPGA. For FPGA-based sequencers, however, it is possible to employ multiple devices to increase the overall amount of the programmed logic. On the other hand, a sequencer based on a processor can perform various readout functions more easily with conditional logic operations. A noticeable overhead time may be added to the total CCD readout time depending on its capability. It must be noted, however, that the operational speed of processors shows a steady increase. For example, Texas Instrument produced the first commercial DSP operating at 5 MIPS(million instructions per second) in 1982. The operational speed of their DSPs increased to 28 MIPS in 1989 and to 1600 MIPS in 1997¹.

Several hybrid sequencers are also found. Bortoletto and D'Alessandro[12] described one based on hard-wired logic circuits and a DSP. The DSP directly controls the generation of clock timing pulses for the vertical transfer whilst hard-wired circuits use a look-up table stored in a dual-port RAM for the horizontal transfer. The sequencer by Bregman and Doorduyn[17] was built with a micro-processor and hard-wired logic circuits. The hard-wired logic circuits generate clock timing pulses for the transfer of the signal charges and the processor controls the order and the number of transfers for the given CCD.

In this work, the sequencer is built with an FPGA, an EEPLD(electrically-erasable programming logic device) to be more precise. It is compact and consumes little power. It is suitable for high-speed CCDs with a minimum overhead time. Moreover, it is in a good accordance with the control method described in the following sub-

¹ <http://www.ti.com/>

section. The prototype low-noise mosaic CCD control system developed at OSL has demonstrated the successful operation of a sequencer based on an EPLD(erasable programming logic device). A reduction of cost and time associated with this work was expected because there were some expertise and facilities from previous work. In addition, the EEPLD development environment provided by the manufacturer makes the sequencer logic more reliable and predictable. The EPM7256E-12² was selected because it supported the highest internal array clock frequency at the time of development and had sufficient internal resources for the sequencer logic.

The majority of CCDs intended for high-speed applications have multiple output nodes as shown in Table [2.2]. The ACCD developed specifically for wavefront sensing has 32 output nodes in spite of its small physical format[64]. It is understood to be an endeavour to reduce the readout noise and to meet the required frame rate at the same time. In the course of the construction of the system hardware, a 64×64 frame-transfer CCD from Loral that has 4 output nodes was provided by the UK Adaptive Optics Programme. Accordingly, the control system was modified to be able to process four output video signals in parallel. If this CCD is read out via all four output nodes, the total readout time significantly decreases approximately by a factor of four. In this case, the required frame rate of 1000 Hz can be easily achieved even with the pixel sampling frequency of 2 MHz. The lower pixel sampling frequency may reduce the readout noise originating from the high-speed operation.

3.2.2 Versatile Control System

There are different high-speed CCDs either commercially available or developed for specific scientific applications. They have their own peculiar optical, electrical, and physical characteristics. Accordingly, each of them requires different operational conditions on voltage levels of biases and clocks, timing sequences for the charge transfers and the supplement circuitry, and widths and overlaps of clock pulses. If a control system has the ability to manipulate these various parameters within pre-defined boundaries, it can be used in many different applications with little modification of its physical circuitry. It will save a considerable amount of resources

² EPM7256E-12 is manufactured by ALTERA. Refer to section 4.4 for its description.

required to design and build a different control system for another specific application. This ability is also critical for prototype or evaluation systems that require the frequent modification of these parameters to find the optimum operational conditions.

Even if the control system in this work was constructed aiming at a specific CCD for wavefront sensing, it was designed to have the versatility to drive other high-speed CCDs that have different electrical and physical characteristics. Two methods adopted in this work to realize a versatile control system are presented in following two subsections.

3.2.2.1 Bitmaps as CCD Clock and Control Timing Pulses

To read out a CCD, the signal charges are transferred to the horizontal register(s) line by line (the vertical transfers). For each vertical transfer, the signal charges in the horizontal register(s) are transferred to the sensing node(s) and sampled pixel by pixel (the horizontal transfers). Two approaches are generally used by control systems in supplying the CCD clock timing pulses to complete a single transfer either vertical or horizontal.

The first approach is the *wait-and-modify* method. In this method, the sequencer stores the pre-defined sequence of the states of clock gates for a single transfer and the wait-numbers assigned to each state. In the beginning, it initialises the clock gates with the first element of the sequence. After counting or waiting for the wait-number assigned to the first element, it retrieves the next element from the sequence and applies it to the clock gates. This process is repeated until the sequence is exhausted, which completes a single transfer either vertical or horizontal.

The second and more frequently used approach is the *bit-slicing* method. In this method, the complete stream of clock timing pulses for a single transfer is stored in memory cells sequentially. The contents in each memory cell define the states of clock gates at an instant. The sequencer provides the sequential access to these memory cells. Therefore, a single transfer is completely represented by the sequential retrieval of the contents from each memory cell. It is called the bit-slicing method because the timing pulse stream is stored in memory cells as if it is sliced by the unit access time to memory cells in the time domain. Control systems with the wait-and-modify

method are found in [18], [61], [29], and [4] while those with the bit-slicing method in [68], [12], [96], and [125].

The work of this thesis employs a typical bit-slicing method that represents the clock timing pulses with a stream of bits. The timing pulse stream for a clock gate is sliced by the period of the master clock running the sequencer. The logical number 1 is assigned to the HIGH clock state and 0 to the LOW state. Therefore, the stream of logical 1's and 0's represents the timing pulse stream. The streams of bits for all clock gates are stored in memory cells collectively. The sequencer provides the sequential access to memory cells for each master clock period. For example, if there are four clock gates as shown in Figure [3.1], a stream of 4-bit wide digital numbers is actually stored.

This collective stream of digital numbers is defined as a stream of Bitmaps or Bitmaps more simply. The number of Bitmap channels or the width of Bitmaps is defined as the number of bits in each digital number. The length of Bitmaps is defined as the number of digital numbers in Bitmaps. The period of the master clock determines the smallest controllable time unit of Bitmaps, which is defined as the resolution unit of Bitmaps. For a specific application, the shape of clock pulses determines the length of Bitmaps for those clock pulses (refer to sub-section 4.6.3) and the required clocking speed does the resolution unit of Bitmaps.

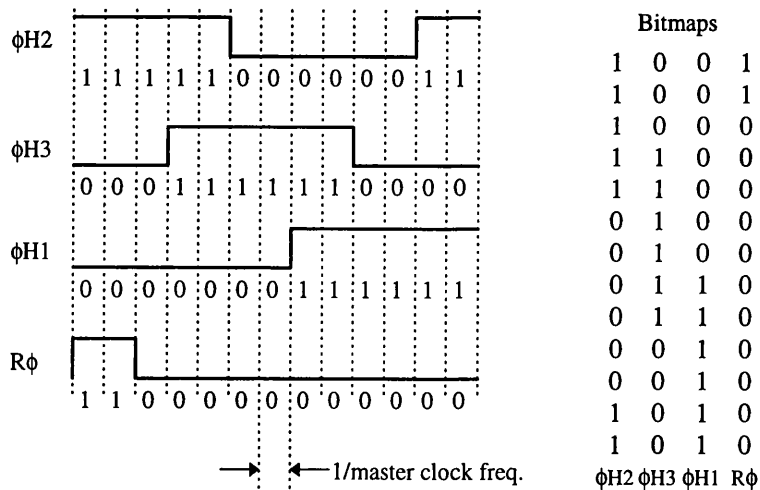


Figure [3.1] A typical stream of Bitmaps to drive four CCD clock gates. The length of this stream of Bitmaps is 13 and its width is 4.

With the Bitmap control method, a set of binary counters can simply constitute the sequencer. For example, if a stream of Bitmaps is m -bit wide and k long, it can be stored in a memory device with the organisation of $n \times m$ bits ($2^n \geq k$) or larger. The sequential access to each memory cell can be performed by a binary counter pre-loaded with $k-1$. For each cycle of the master clock, it is down-counted (or up-counted depending on the actual logic implementation) whose output bus is directly mapped onto the address bus to the memory device. This relationship is schematically shown in Figure [3.2] for the Bitmaps given in Figure [3.1].

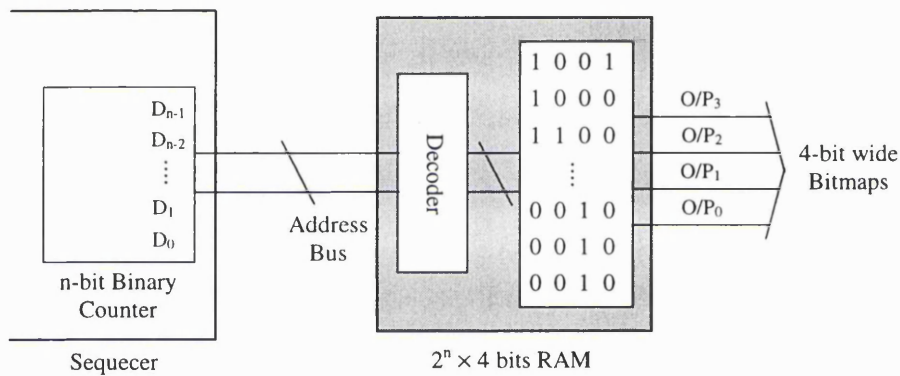


Figure [3.2] Relationship between the binary counter and the external memory device in the Bitmap control method

Furthermore, this idea can be applied in controlling the numbers of transfers needed to complete the readout of a single image frame. All CCDs have the common characteristic that each vertical transfer is followed by a number of horizontal ones in the readout procedure. The physical format of a CCD determines the required numbers of vertical, horizontal, or frame transfers if necessary. Several binary counters in the sequencer are pre-loaded with these numbers as in the case of Bitmaps. The readout of an image frame is performed by counting up or down these binary counters in an appropriate serial order. The combination of their carry-out's is also useful to select the correct Bitmaps for the next transfer. In this manner, the sequencer logic tends to be register-intensive, which is ideal for an FPGA(s).

The distinctive feature of the Bitmap control method is that the control timing pulses can be part of Bitmaps. For example, they may include those for sampling the video signals, analog-to-digital conversion, and storage of the sampled signals. Its usefulness and efficiency was demonstrated by Han[41] with the prototype mosaic CCD control system. With two additional bits in Bitmaps, he defined streams of clock and control timing pulses to drive a mosaic of four CCDs in a multiplexed way. If not limited by the available amount of memory cells, the binning of pixels in any degree can be easily performed. The new Bitmaps constructed by adding two horizontal transfer Bitmaps can perform the horizontal binning by a factor of two, for example. Apparently, the appropriate modification of the streams of control timing pulses is necessary when two Bitmaps are added.

One of the disadvantages common to all bit-slicing sequencers is that the longer Bitmaps are or the smaller the Bitmap resolution unit is, the more memory cells are needed. As pointed out previously, the difficulty to implement complex conditional logics in an FPGA(s) and the nature of the bit-slicing method limits the sequencer to perform sophisticated readout functions such as the readout of a sub-section of a multiport CCD.

If the modification of Bitmaps and the numbers of transfers is allowed to the user, the sequencer can generate the streams of clock timing pulses virtually for any CCD. This feature is also very useful in finding the most efficient clock pulse streams to drive a specific CCD, i.e., the optimisation of the streams of clock timing pulses. In the work of this thesis, the user can easily define and modify these system parameters through the user-interface. Therefore, the control system described in this thesis can drive CCDs of more than 100×100 pixels without any physical modification and produce the spatial resolution high enough for wavefront sensing in visible bands. To the best of the author's knowledge, there is no sequencer for high-speed CCDs that is based on an FPGA(s) and has this versatility. The most comparable sequencer in terms of versatility would be the one by Yates *et al.*[125]. They used the programmable digital word generator.

3.2.2.2 Static DAC Control

To drive CCD clock gates, the clock timing pulses from the sequencer must be converted to have appropriate voltage levels ($\pm 5 \sim \pm 20$ V). The CCD also requires the provision of adequate biases. One possible approach is to define the clock and bias voltage levels using potentiometers driven from fixed power supplies or in similar manners. However, in the process of searching for their optimum values, this approach is inefficient and prone to human errors because frequent readjustments are necessary. Another simple, but powerful approach is to utilise DACs (digital-to-analog converters) to define these voltage levels, which are controllable by the system software.

There are two possible ways to make use of these DACs; static and dynamic DAC control methods. In the static method, two independent DACs are employed to define two clock voltage levels, HIGH and LOW, for each clock gate. They maintain their initial values throughout the readout procedure. Unlike in the former method, DACs are updated during the readout in the dynamic method. It is evident that one DAC is sufficient for each clock gate.

The static method is simpler and is in good harmony with the Bitmap control method because each channel of Bitmaps can directly control a specific clock driver that switches between two voltage levels. Figure [3.3] illustrates a simple clock driver driven by a Bitmap channel with the static method. With the dynamic method, on the other hand, the control system needs additional logic circuits to map the Bitmaps onto DAC values and to update DACs. The characteristic of DACs in terms of the setup time is critical to the high-speed operation in this case. However, it can be more instrumental when a more complex CCD clocking is desired such as variable clock voltage levels between vertical and horizontal transfers or the tri-level clocking for MPP CCDs. In this work, the static DAC control method is employed to construct simple and effective CCD clock drivers.

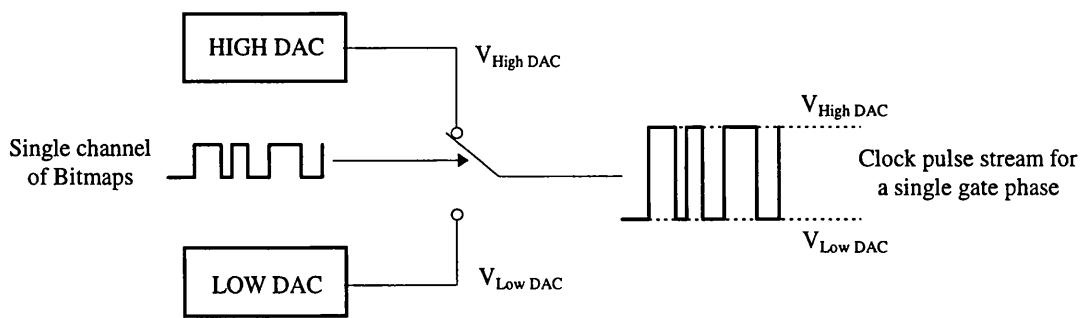


Figure [3.3] Provision of clock pulses with the static DAC and Bitmap control methods

3.2.3 Modular and Compact System with the User-Friendly Interface

A well-designed modular system ensures a concrete conceptual understanding and rapid progress in its development and evaluation phase. The frequent repetition of the modification and evaluation process for a prototype system also demands it to be made of well-defined modules. It ensures rapid modification when necessary by isolating each module with its inputs and outputs defined. It saves resources not only when a prototype system is initially developed but also when its upgrade is necessary.

In this work, the system hardware was designed to be modular to the lowest level possible. It consists of several modules that are defined according to their distinctive functionality. Each module is represented by its board name as described in detail in Chapter 4. The sequencer logic is also made up with small function modules. Components on each hardware module board were grouped in the layout planning when it does not impede its performance. The control software reflects the design ideas more evidently than the hardware. Modular and well-structured control software routines are, therefore, critical to a system in the course of continuous development and upgrade by someone other than the original designer. All source codes for the user-interface and control software were written with the object-oriented programming language C++, which inherently demands a modular and top-down design. The control of the system is performed through various modular class objects. The structure of the user-interface and control software and class objects are described in Chapter 5 in detail.

As the computing power and physical density of modern electronic systems increases exponentially, much endeavour has been put into the development of electronic devices consuming less power and occupying less space. Because the high-speed CCD control system involves components operating at high frequency and CCD cells are inherently capacitive, the number of high-speed components dictates the amount of power consumption and thus the heat generation. They include those employed to generate the clock timing pulses and to drive the CCD clock gates. A sequencer based either on an FPGA or on a processor, therefore, may contribute to the reduction of the number of components and the power consumption. In this work, the sequencer logic is integrated in an FPGA. An AT-compatible PC was used as the development platform. It controls the system and serves as the medium for user-interface and image display. With the main power supply unit and hardware modules integrated on a sub-rack as described in Chapter 6, it constitutes a compact, portable, and stand-alone control system.

It is usually found that the user experiences difficulty in managing and manipulating many system parameters provided by a sophisticated control system. A user-interface that provides an easy access to these parameters is essential, especially to a system that requires their frequent modification for optimisation. The high-speed CCD control system incorporates a graphical user-interface(GUI) to support its versatility. In a GUI, the user is generally presented a kind of artificial environment, an abstraction of the physical one. In this work, the GUI grants a user the full control of all system parameters in a window-like control environment. For instance, they include all DAC values, the numbers of transfers, and the readout mode. It also incorporates a graphical editor that represents and manipulates the Bitmaps visually in the form of pulse waveforms. It reduces the possibility that the user makes critical errors in defining the Bitmaps such as the overlaps of clock pulses. This editor provides the Bitmaps in conventional text files that can be directly utilised by the sequencer to generate the clock timing pulses.

3.3 Other High-Speed CCD Imaging Systems

3.3.1 Motion Analysis and High-Speed Image Acquisition

With respect to the readout speed of CCD imaging systems, the most demanding application is motion analysis and high-speed image acquisition. This field of application includes analysis of mechanical dynamics and fluid flow, combustion analysis, analysis of the ballistics of munitions, and development of medical equipment, to name a few. To compete with the high-speed photography that has been widely used in this field, Hughes[48] stated that CCD or MOS imaging cameras must produce image formats larger than 1000×1000 pixels and frame rates between 150 and 1000 frames per second. Etoh and Takehara[30] carried out an informative survey in the Japanese scientific and engineering community to compile information on the applications of the high-speed videography and their speed requirements. Their survey shows that high-speed imaging cameras with frame rates up to 10^3 to 10^4 frames per second cover only 30 to 40 % of the potential applications, whereas those of 10^6 to 10^7 frame per second cover more than 90 %. The bar chart that illustrates the frame rate requirement is reproduced in Figure [3.4]. It must be pointed out that this requirement is only applicable to burst readout mode.

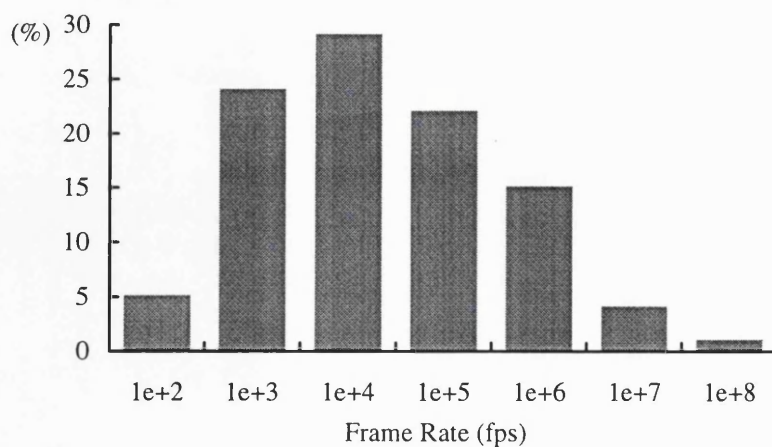


Figure [3.4] Required frame rate for the high-speed videography reproduced from Etoh and Takehara[30]

Many CCD imaging systems were developed for motion analysis using either conventional or specific high-speed CCDs. The motives to use conventional CCDs are the high price of customised high-speed CCD cameras and their relatively low dynamic range imposed by the capability of control electronics and the high readout noise. The feasibility to use conventional CCDs is highly dependent upon the characteristics of the imaging targets and their environment.

Hiller and Kowalewski[45] used a frame-transfer CCD for the analysis of liquid jets in two specific cases. In the case of multi-exposure during integration, a moving target whose images are superimposed onto the CCD image section is imaged several times during integration. It is necessary to adjust the illumination level of the light source to ensure that a single exposure generates electrons above the noise and that multiple exposures generate electrons below the full well capacity. In the other case of multi-exposure during frame transfer, only a strip of pixels is imaged and their signal charges are transferred into the CCD storage section for the next exposure. Consequently, a temporal sequence of images spatially shifted relative to each other is stored in the storage section. This method is appropriate to imaging targets of oblong shape such as liquid jets in high speed.

A similar method was tried by West *et al.*[123] with a slow-scan full frame CCD. They constructed a high frame-rate burst imaging system with performance comparable to slow-scan cameras by using different rates for imaging and readout. They acquired shifted images from a strip of pixels out of 1152×298 total pixels at the frame rate of 2 kHz. Other pixels were optically masked. For higher frame rates, they also suggested a customised fibre-optic image reorganiser to map the target image onto a strip of a larger CCD.

Racca *et al.*[92] took a different approach with conventional CCD cameras in motion analysis. They multiplexed the target image temporally onto several independent CCD cameras with beam splitters and fast liquid crystal shutters. These CCDs are read out sequentially at the conventional frame rate. A multiplexed readout of CCDs can also be employed with less complex electronics. This system produces images of a higher vertical spatial resolution compared with those that use a strip of pixels.

Lula *et al.*[69] developed a high-speed CCD imaging system for high-resolution real-time optical inspection of part surfaces required in the semiconductor industry. They used a 256×128 interline-transfer CCD that has 64 output nodes to achieve the frame rate of 23000 frames per second. Lawler *et al.*[60] also used a 512×512 high-speed frame-transfer CCD with 16 output nodes to construct an image acquisition system with the readout speed up to 500 frames per second. The CCD has the on-chip three-stage CDS(correlated double sampling) amplifier for each output node, which significantly contributes to the reduction of the number of off-chip components and the size of the overall system. They used a circular buffer memory to store up to 256 images in the burst readout mode. Levine *et al.*[65] used a very similar frame-transfer CCD for their high-speed imaging system, which has 16 output nodes and 16 on-chip CDS amplifiers. Prokop *et al.*[90] elaborated an imaging camera for the recording from oscilloscopes and framing cameras within 1 ms of the experimental event with a multiport 1024×256 CCD. The CCD also has 16 output nodes and was designed to have 2-phase clock gates to enhance the readout speed. Another example is the High Speed Solid State Imager Test Station developed to evaluate high-speed multiport CCDs by Yates *et al.*[125]. It has a pixel sampling rate of 10 to 100 MHz. It was used to evaluate two multiport CCDs, CCD-13 and HS0512J manufactured by EEV and Reticon, respectively. Table [3.1] is reproduced from Hughes[48] with the addition of several other CCD cameras to show some of commercially available high-speed CCD/MOS cameras.

3.3.2 High-Speed Photometry

Another application that requires relatively high-speed CCD imaging systems in astronomy is 2-dimensional photometry. Abbott and Opal[2] stated that it is necessary to measure the variations in brightness of astronomical objects on time scales ranging from several hours to milliseconds for high-speed photometry. They demonstrated a high-speed photometer on 0.8 m and 2.1 m telescopes at the McDonald Observatory. It employs two frame-transfer CCDs in order to reduce the time gap between observations and thus, to increase the frame rate. Stover[107] also used a frame-transfer CCD to construct a high-speed photometer with the readout time of 9

seconds. Considering that high-speed photometry requires CCDs of larger formats than motion analysis and high-speed image acquisition, continuous readout mode rather than the burst mode, and more stringent performance in terms of dynamic range and readout noise, it is evident that the development of an efficient high-speed CCD control system is beneficial.

3.3.3 Soft X-ray Imaging

High-speed CCD imaging systems also find many applications in the field of the synchrotron radiation research for the direct detection of soft x-rays. In addition to the optimisation of numerous parameters of the CCD itself to achieve enhanced QE in the energy band of 1 to 10 keV, the imaging system needs to have high dynamic range, high-speed data handling capability, and high spatial resolution. Rodricks *et al.*[97] developed a large-area x-ray imaging system for synchrotron applications by mosaicking four 584×390 CCDs. These four CCDs are coupled to a fibre-optic taper and image intensifiers and read out in parallel in less than 1 second. They used memory devices to store up to eight image frames that are transferred to a hard disk at the completion of the image acquisition. They emphasised that the high-speed operation is especially crucial for experiments in structural biology and protein crystallography where samples deteriorate rapidly under the high radiation exposure.

To record the diffraction pattern from an epitaxial layer of Ge grown on GaAs in real time, Clarke[24] used a 512×512 MPP CCD. To resolve the diffraction pattern as a function of time with the 1 MHz frame rate, he used a single row of the CCD for imaging. The system gives temporally resolved diffraction patterns subsequently stored in CCD rows. This is a very similar approach to those exploited by many other systems described in sub-section 3.2.1.

Mfg.	Model	Max. Frame Rate (fps)	Format	Video Signal Sampling	Data Rate (Mbit/s)
Sarnoff	SAR 1024	210/300	1024×1024	12 bit	2642
Kodak	1000HRC	1000	640×480	8 bit 1-chip colour	2458
Kodak	Ektapro 4540	4500	256×256	8 bit	2359
Vision Research	Phantom	500	512×512	8 bit	1049
NAC	RX-1	500	580×434	8 bit 1-chip colour	1007
Princeton Instruments	HSI/CCD-1152	2000	298×100	14 bit	834
Sarnoff	VCCD512	200/400	512×512	12 bit	629
AOA ³	MD4256C	1000	256×256	8 bit	524
Kodak	Ektapro Hi-spec	1000	192×239	8 bit	367
Dalsa	CA-D4-1024	40	1024×1024	8 bit	336
Lake Image Systems	MotionScope 500	500	336×243	8 bit	327
EG&G Reticon	MC4013 MD4013	30	1024×1024	8 bit	252
Dalsa	CA-D1-0256	220	256×256	8 bit	115
Dalsa	CA-D2-0512	53	512×512	8 bit	111
Kodak	Ektapro EMP	250	192×239	8 bit	92
EG&G Reticon	MC9000	105	256×256	8 bit	55
Loral Fairchild	CAM4002	300	256×256	Raw Video	
JC Lab	HSC-500	500	510×246	NA	

Table [3.3] List of some commercially available CCD/MOS cameras with high frame rates in ascending order of data rate[1, 20, 49] (originally from Hughes[48])

The survey was conducted at the beginning of 1996. The information on some cameras are from data sheets that are not given in Bibliography.

³ Adaptive Optics Associates, Inc.

Prototype Control System Hardware

The performance of the control system must be evaluated with a specific CCD even if it is designed to have the versatility to drive virtually any high-speed CCD within the limits. Initially, the prototype system hardware was designed with the assumption that the CCD has the format of 100×100 pixels and a single output node. It gives a pixel sampling frequency of approximately 10 MHz for the frame rate of 1000 Hz. A crystal oscillator of 80 MHz was chosen as the master clock of the sequencer to satisfy the pixel sampling frequency. With the Bitmap control method, therefore, the resolution unit of Bitmaps is 12.5 ns. In the course of its construction, the UK Adaptive Optics Programme provided one of the Loral 64×64 frame-transfer CCDs for the work of this thesis, which led to the completion of the prototype hardware based on it.

To read out a frame-transfer CCD (see Figure [4.3] for the architecture of a typical frame-transfer CCD), the signal charges are transferred from the image section to the storage section(s) (frame transfer). The signal charges in the storage section(s) are transferred to the horizontal register(s) line by line (vertical transfer). Finally, the signal charges in the horizontal register(s) are read out through the output node(s) pixel by pixel (horizontal transfer). Three fundamental streams of Bitmaps are defined to drive the Loral frame-transfer CCD; frame-transfer, vertical transfer, and horizontal transfer Bitmaps. In addition, another stream of purge Bitmaps is defined to support the purging of the CCD. Purging refers to clearing out the signal charges accumulated in pixels at high-speed without sampling the output video signal.

The frame-transfer Bitmaps are defined as those to transfer a single line of signal charges from the image to the storage sections. Therefore, to complete the frame transfer, the generation of the frame-transfer Bitmaps must be repeated. The vertical transfer Bitmaps are defined as those to transfer a single line of signal charges from the storage sections to the horizontal registers and the horizontal transfer Bitmaps as those to sample a single pixel in the horizontal registers. The sequence of Bitmap generation to complete the readout of an image frame from a frame-transfer CCD is depicted in Figure [4.1] when the purging is disabled.

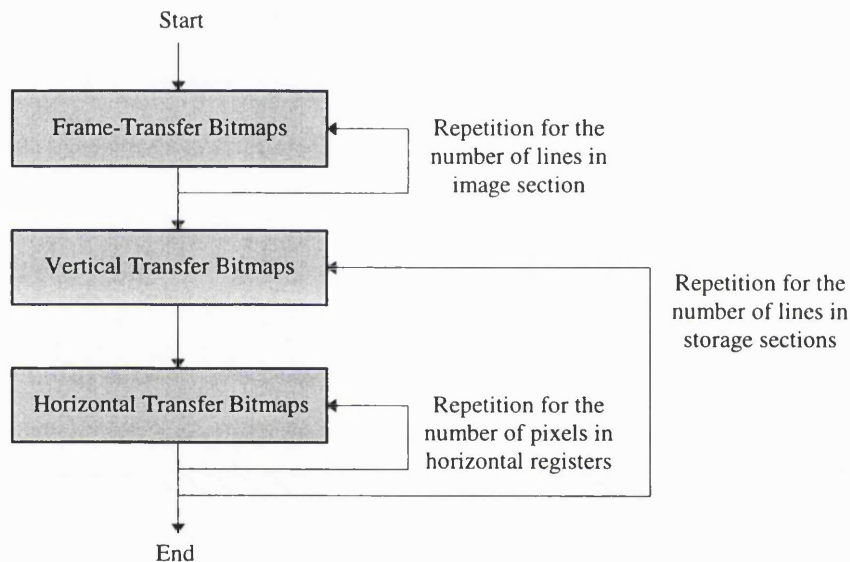


Figure [4.1] Sequential generation of the Bitmaps to acquire an image frame from a frame-transfer CCD with the purging disabled

Nine functional modules that constitute the prototype hardware of the control system along with the host PC are listed below. Figure [4.2] is the block diagram of the prototype system hardware. As its name implies, the main power supply unit provides power supplies to each module, which are applied either directly or after regulation by the Voltage Regulation Board. The Bitmap Generation Board is responsible for the generation of Bitmaps with the sequencer, the control of DACs on Bias and Clock Boards, and other miscellaneous functions. The Bias Board supplies constant biases to the CCD and two Clock Boards generate clock pulse streams to read out the CCD

from the given Bitmaps. Output video signals from the CCD are processed and stored by two Signal Processing Boards. The overall control is performed by the host PC through an I/O interface card. The CCD resides on the CCD Board, which is application-dependent. The more detailed description of the target CCD, the Loral 64 $\times$ 64 frame-transfer CCD, and each module is made in the following sections except for the CCD Board, which is presented in section 6.1.

- Main Power Supply Unit and Voltage Regulation Board
- Bitmap Generation Board
- Bias Board
- Two Clock Boards
- Two Signal Processing Boards
- CCD Board

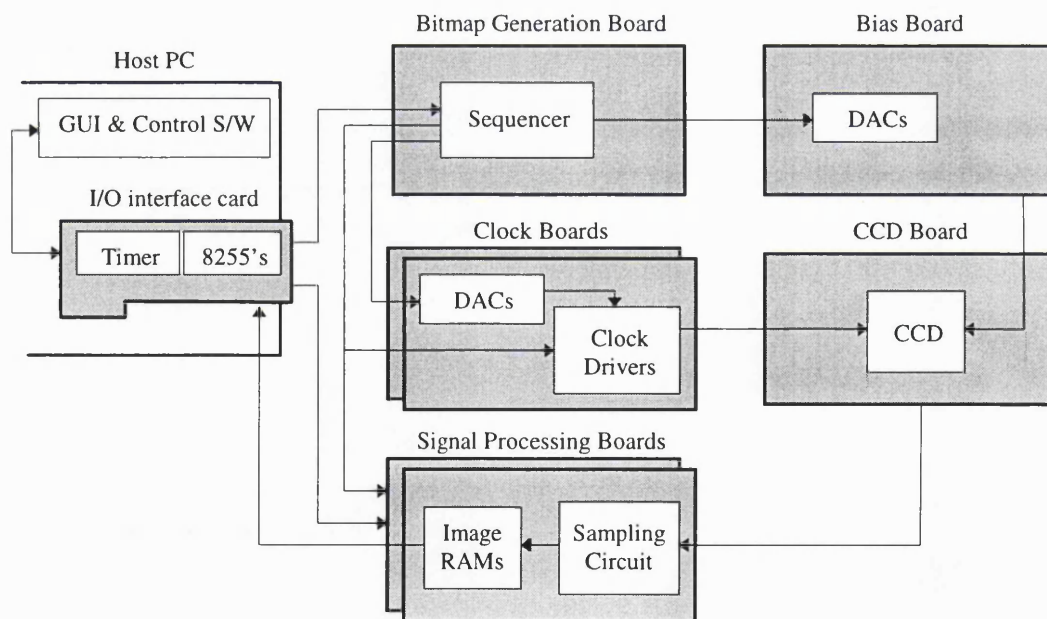


Figure [4.2] Block diagram of the prototype hardware of the high-speed CCD control system

4.1 Loral 64×64 Frame-Transfer CCD

The 64×64 frame-transfer CCD is one of a batch of 32 CCDs procured from Loral and it originated from a wafer made for Dr. Johannes Anderson at the Copenhagen Observatory. The CCD was made available through the kind cooperation of Dr. Craig MacKay in the University of Cambridge. The CCD is packaged in a 68-pin PGA and is organised as a 64×64 image section and two 32×64 storage sections. It has a horizontal register at each end of its storage sections and two output amplifiers at both ends of each horizontal register. The format and pin definition of the CCD is illustrated in Figure [4.3].

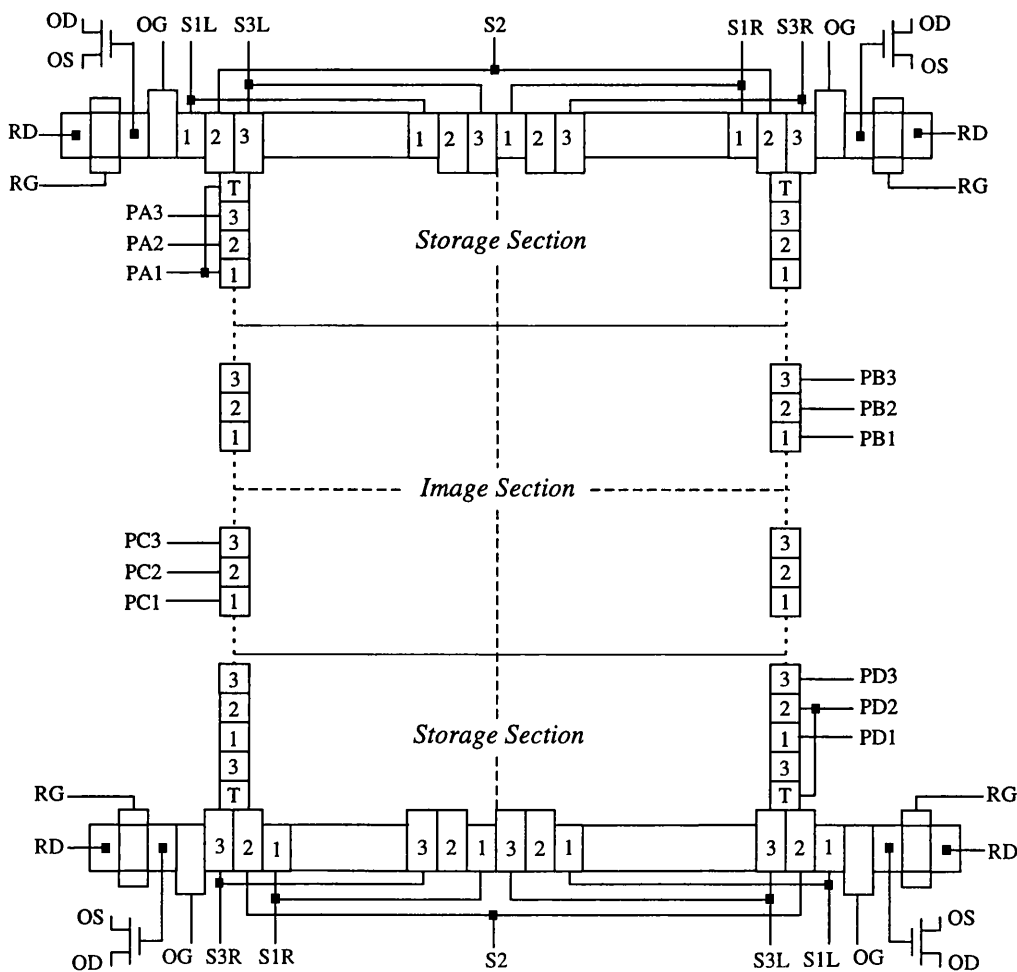


Figure [4.3] Format and pin definition of the Loral 64×64 frame-transfer CCD

MacKay[73] tested one of these CCDs as a single 128×64 full frame device in a compact liquid nitrogen dewar within the temperature range of -40 to -60 °C. In this test, most CCD parameters were optimised to result in the optimum performance. Some of its principal characteristics measured in this test are summarised below.

- *Full-well capacity is approximately 10^4 electrons.*
- *CTE is good when measured at low light levels with signals in 15 to 25 electrons peak level.*
- *Detective QE showed its peak value of 0.59 at 690 nm out of six wavelengths between 470 and 940 nm. In another set of test results, the highest QE was 44.5 % at 700 nm out of 14 wavelengths between 400 and 1060 nm[74].*
- *Dark current was measured to be about 0.4 electrons/pixel/second at -56 °C and about 3 electrons/pixel/second at -40 °C.*

4.2 Host PC

An AT-compatible PC (host PC) was used as the development platform. The control software performs the control of the system hardware through an I/O interface card resident on one of its ISA slots. The interface card has two 8255 devices and two separate timers on board. Outputs from two 8255's are grouped to give two independent 16-bit wide data buses that can be further divided into two separate byte wide channels. Each byte wide channel can be configured independently in various read or write modes by manipulating the onboard control registers.

One of two 16-bit wide data buses is used to transmit various sets of data and control signals to the sequencer on the Bitmap Generation Board and the Image RAM Addresser on the Signal Processing Board #1. The transmission is accomplished in the strobe mode. The active LOW OBF(output-buffer-full) is used as the strobe signal, which is reflected on the Bitmap Generation Board back to the interface card to supply the active LOW ACK(acknowledge). The retrieval of the sampled pixel values from the Image RAMs and their transmission to the host PC is also accomplished in the strobe mode through a half of the other 16-bit wide data bus. In this instance, the active LOW

STB(strobe) is reflected on the Signal Processing Board #1 to provide the active LOW IBF(input-buffer-full).

Generally, frame-transfer CCDs do not require any shutter mechanism because the storage section(s) is opaque. The integration (or exposure) time is normally given by the readout time taken to sample the signal charges stored in their storage section(s). When an integration time shorter than the readout time is desired, it can be achieved by manipulating the states of vertical clock gates of the image section. However, because the Loral frame-transfer CCD is an MPP device that has the capability to integrate the signal charges with all vertical clock gates at LOW states, the readout time is the only factor that governs the integration time unless a shutter mechanism is employed.

Initially, the sequencer was designed to have a logic block that performed the manipulation of a shutter mechanism when necessary after the completion of frame transfer. However, the logic block required a large amount of resources to produce the integration time in the order of hundreds of μ s with the master clock of 80 MHz. It made the fitting of the sequencer logic into a single EEPLD impossible. The required amount of resources can be reduced if the logic block is driven at lower frequencies through a series of T-flipflops.

Instead, the host PC performs the manipulation of a shutter mechanism using one of two timers on the interface card. The counter in the timer is loaded with a DN(digital number) in accordance with the given integration time. The count-down is commenced by the end-of-frame-transfer signal from the sequencer. The end-of-frame-transfer signal and the carry-out signal at the end of count-down are used by the sequencer to generate a square pulse. The width of the square pulse is equal to the DN multiplied by the timer clock period(see sub-section 4.4.2.6). Therefore, the square pulse is useful to define the integration time for frame-transfer CCDs.

For full-frame CCDs, the signal charges are read out after integration. In this case, the frame-transfer Bitmaps can be used to manipulate a shutter mechanism. The maximum integration time obtainable in this way is the maximum number of repetitions of the frame-transfer Bitmaps times the maximum Bitmap length multiplied by the Bitmap resolution unit. Refer to section 6.4 for an example where

the integration time is controlled in this manner. If no Bitmap channel is available or the maximum integration time achievable is too short, a different method must be used.

4.3 Main Power Supply Unit and Voltage Regulation Board

The power supply unit used in the development of the mosaic CCD control system was modified and renovated for the work of this thesis. Several old supply modules were replaced by new ones that supply higher currents. The main power supply unit consists of seven DC linear power supply modules to provide analog voltages of ± 23 , ± 15 , ± 5 V, and the digital voltage of $+5$ V to module boards. Except for ± 23 V supplies that are regulated to levels in ± 18 to ± 20 V by two adjustable linear voltage regulators on the Voltage Regulation Board, others are filtered by ferrite beads and capacitors and relayed directly to each module board. It was tried to use power and ground cables dedicated to each module board. However, no suitable power bus bar was available. Alternatively, two bus bars are used; one for analog power supplies and ground and the other for digital supplies and ground (see section 6.3). The power supplies needed for each module board are summarised below. These figures originate from the operating conditions of devices used and some scientific CCDs.

- Bitmap Generation Board : $+5$ V digital
- Bias Board : $\pm 18 \sim 20$, $+15$ V analog, and $+5$ V digital
- Clock Boards : ± 15 V analog and $+5$ V digital
- Signal Processing Boards : ± 5 V analog and $+5$ V digital
- CCD Board : $+18 \sim 20$ and ± 5 V analog

4.4 Bitmap Generation Board

4.4.1 Overview

The Bitmap Generation Board is the heart of the system hardware. Its primary function is to supply the streams of CCD clock and control timing pulses, i.e., the Bitmaps to read out the CCD in accordance with its physical format. They are

transmitted to two Clock Boards and two Signal Processing Boards. It is also responsible for the control of octal DACs on Bias and Clock Boards, which are used to define the bias and clock voltage levels. The Bitmap Generation Board consists of an 80 MHz crystal oscillator (the master clock), eighteen opto-isolators, six high-speed quad differential line drivers, three octal buffers, and the sequencer with three high-speed static RAMs. These RAMs are used to store the Bitmaps and referred to as Bitmap RAMs. The layout of the Bitmap Generation Board is shown in Figure [4.4].

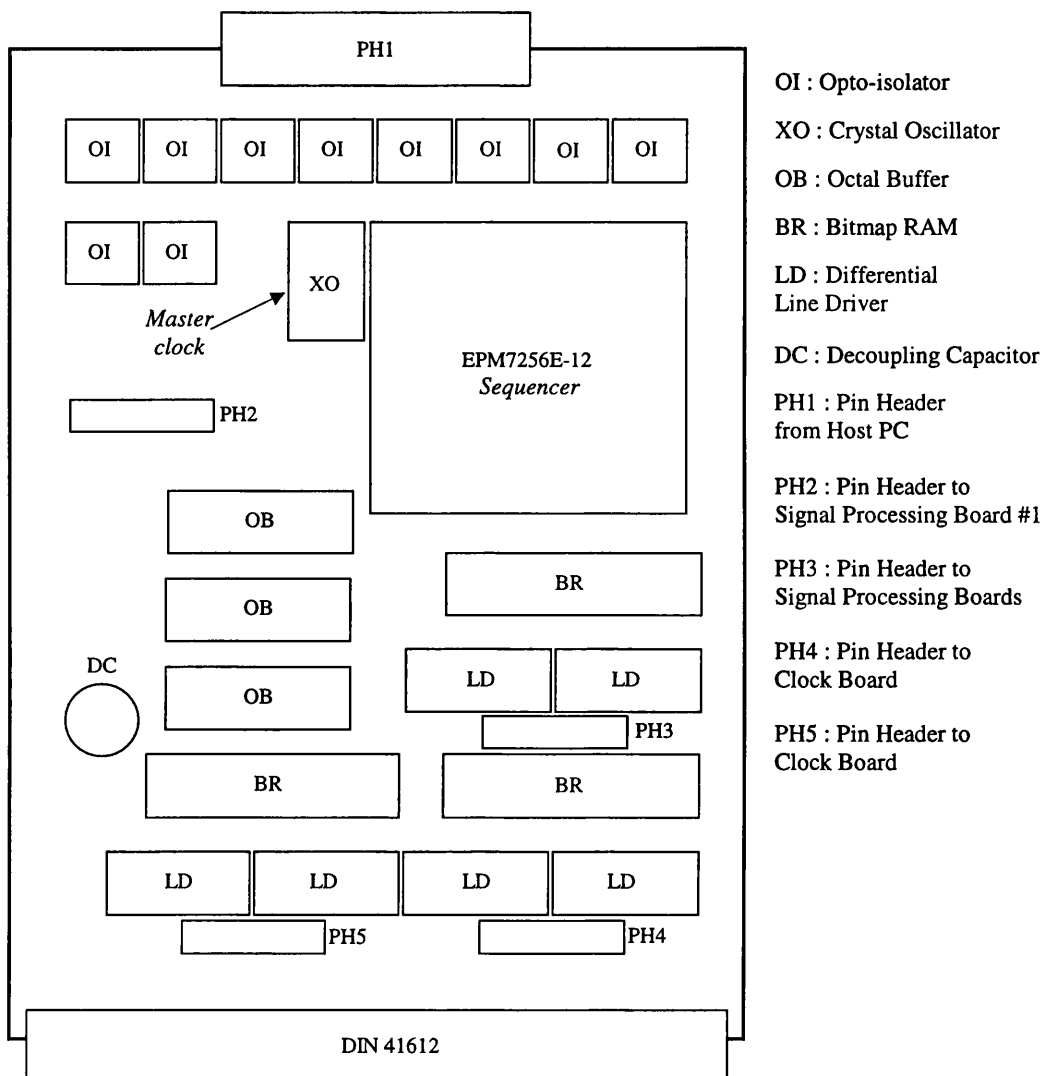


Figure [4.4] Layout of the Bitmap Generation Board
See Figure [4.5] for the organisation of Bitmap RAMs and their mapping by the sequencer.

4.4.2 Sequencer, EPM7256E-12

Physically, the sequencer is an EEPLD, EPM7256E-12, which is programmed with the logic that embodies the Bitmap and static DAC control methods. Strictly speaking, the sequencer consists of the EPM7256E-12 and three Bitmap RAMs because the Bitmaps are stored in and retrieved from them. Currently, there are three leading companies in the FPGA market; Altera, Xilinx, and Actel. Altera produces a large number of devices based on the traditional EPLD technology and Xilinx shows strong competition in the FPGA market based on the static RAM logic cell architecture. Actel has pioneered the antifuse technology to produce devices with high tolerance in harsh environments. The EPM7256E-12 is one of the MAX(Multiple Array matrix) 7000E devices, which are higher density members of the MAX 7000 family manufactured by Altera.

MAX 7000 EPLDs use CMOS(complementary MOS) EEPROM(electrically-erasable programming read-only memory) cells to implement logic functions. The user-configurable MAX 7000 architecture can accommodate a variety of independent combinational and sequential logic functions. The EPM7256E-12 provides 256 advanced macrocells (or logic cells) combined into 16 groups called LABs(Logic Array Blocks) to efficiently implement registered and complex combinational logic. It has a maximum internal array clock frequency of 90.9 MHz, which gives the minimum array clock period of 11 ns. The EPM7256E-12 was selected as the sequencer because it had the highest internal array clock frequency and sufficient resources at the time of development.

The development package, MAX+PLUS II provides an integrated programmable logic design environment to Altera devices. Generally, the design and programming of a device follows four phases; design entry, design compilation, design verification, and device programming. The MAX+PLUS II provides various design entries such as graphic, text, and waveform. In the phase of design compilation, it performs functions such as logic synthesis and design fitting and produces various files for programming, simulation, and timing analysis. With the simulation tool, the designer can define the inputs to the device and examine its outputs or registers' contents in waveform style.

The designer can also examine the timing delay between logic cells with the timing analysis tool.

4.4.2.1 Sequencer Logic

The sequencer was programmed with logic functions to control the octal DACs on Bias and Clock Boards, to store four streams of Bitmaps defined by the user in Bitmap RAMs, and to read out the Loral 64×64 frame-transfer CCD with the given Bitmaps. Its logic consists of several function blocks whose brief description is given below. Their functional description is presented in the following sub-sections. Appendix A shows the input and output pin assignment of the sequencer, the schematic block diagram at the top level of its logic, and the schematic block diagram of BM_CNT block.

- INTERFACE block decodes 5 LSBs(least significant bits) of the 16-bit wide data bus from the interface card on the host PC to generate various control and write ENABLEs for other function blocks. Other bits are used to represent DNs to be stored in latches. It also interfaces the master clock input and the OBF signal to other blocks.
- DAC_CTRL block decodes 8 LSBs out of the data bus from the interface card to provide write ENABLEs and the address bus to octal DACs on Bias and Clock Boards.
- CLK_CTRL block generates the M_CLK signal that serves as the clock to the counter in the BM_CNT block. It originates from the OBF signal in the writing procedure of the sequencer and from the master clock in the CCD readout procedure. The CLK_CTRL block also controls two signals, READING and WRITING that are used to enable many registers.
- WR_CTRL block manipulates two control signals, SEL_HI and SEL_LO to ensure that the correct storage block of a Bitmap RAM is addressed in the writing procedure of the sequencer. It also generates the WR_END signal when the storage of four sets of Bitmaps into a Bitmap RAM is completed.
- RD_CTRL block manipulates two control signals, SEL_HI and SEL_LO to ensure that the correct storage block of a Bitmap RAM is addressed in the CCD readout procedure. The carry-outs from the counters in FT_CNT, LT_CNT, and

HT_CNT blocks are used for their manipulation. It provides a counter to support two different readout modes; burst and continuous. In the burst mode, the number of image frames to be acquired is stored in the counter and the REAL_END signal is generated when the readout is completed. In the continuous mode, image frames are acquired until the user terminates the readout procedure. In both modes, it generates the RD_END signal at the acquisition of one image frame.

- RAM_CTRL block provides output and write ENABLEs to Bitmap RAMs using the master clock and the OBF signal. All write ENABLEs are synchronised to the master clock. It also generates the RAM_CLK signal that is used by the MIS_CTRL block to generate two MSBs (most significant bits) of the Bitmap RAM address bus.
- MIS_CTRL block generates other miscellaneous control signals, such as PURGE, RST, and M_RST. It also generates two MSBs of the address bus to Bitmap RAMs using OR-gated SEL_HIs and SEL_LOs from WR_CTRL and RD_CTRL blocks in both writing and readout procedures.
- FT_CNT block has latches to store the number of repetitions of frame-transfer Bitmaps. Its counter is loaded with this number when the readout of a single image frame is initiated and counted down when each occurrence of frame-transfer Bitmaps is completed. The FT_END signal is generated from its carry-out at the end of the count-down.
- LT_CNT block has latches to store the number of vertical transfers. Its counter is loaded with this number when the readout of a single image frame is initiated and counted down when the horizontal transfer is completed for the given number of times. Its carry-out, LT_END is generated at the end of the count-down.
- HT_CNT block has latches to store the number of horizontal transfers. Its counter is loaded with this number when the vertical transfer of pixels is completed and counted down when each occurrence of horizontal transfer Bitmaps is completed. Its carry-out, HT_END is generated at the end of the count-down.

- BM_CNT block provides four groups of latches to store the numbers of resolution units of four Bitmaps. Using OR-gated SEL_HIs and SEL_LOs from WR_CTRL and RD_CTRL blocks, it loads the counter named the Bitmap counter with the appropriate number of resolution units. The counter is counted down by the M_CLK signal, the output bus of which defines the address bus to three Bitmap RAMs except for two MSBs. It provides the carry-out, M_END and the look-ahead end-of-count-down signal, AHEAD_END.
- SHUTTER block generates a square pulse, SHUTTER using FT_END from the FT_CNT block and INTG_END from the interface card to define the integration time.

Because the propagation delay through logic primitives such as MCELL depends on the operating conditions like the temperature, their use for the generation of time delay in Altera devices can cause the ‘race condition’ between logic circuits and lead to the unreliable operation[27]. To prevent it in the sequencer logic, either the master clock or the OBF signal was used for timing and synchronisation wherever possible instead of logic primitives. It apparently requires several additional registers or flipflops. An expander in Altera devices is an AND gate logic used to generate the combinational logics for INPUT, CLOCK, and CLEAR to flipflops. Because the sequencer logic requires a large amount of interconnects between logic cells, any logic primitives implemented with expanders were also avoided.

The operation speed of the sequencer was decided based upon the author’s past experience with Altera EPLD devices. In the previous low-noise mosaic CCD control system, an EPM5128 from the Altera MAX 5000 family was employed as the sequencer. After its logic was modified to remove unnecessary logic primitives, the EPM5128 was driven at the highest internal array clock frequency of 33.3 MHz specified in the data book. During its simulation and testing, it was observed that it exhibited satisfactory outputs when driven at frequencies less than 0.9 of the maximum internal array clock frequency specified. When driven at frequencies close to its maximum, its outputs showed unexpected behaviour, which is believed to have been caused by the propagation delay between logic cells and the set-up and hold times in each of them.

The simulation result with the EPM7256E-12 also revealed that its outputs exhibit the expected behaviour only when driven at frequencies less than 0.9 of the maximum internal array clock frequency specified. In the readout procedure of the CCD, the sequencer is driven by the master clock of 80 MHz to generate Bitmaps, which is close to 0.9 of the maximum internal array clock frequency of 90.9 MHz¹. It gives the Bitmap resolution unit of 12.5 ns. Therefore, eight resolution units are assigned to the horizontal transfer Bitmaps for the pixel sampling frequency of 10 MHz (100 ns). Eight resolution units are sufficient to define the horizontal transfer Bitmaps precisely with the clock pulse overlaps.

In the compilation phase of the MAX+PLUS II development package, the fitting process optimises the designed logic, decides the required amount of resources such as macrocells and interconnects, and attempts to allocate resources provided by the specified device accordingly. Even with the large amount of resources available in the EPM7256E-12, much difficulty was experienced in fitting the sequencer logic because of its register-intensive nature. After several failures in this process, a substantial amount of effort was put into the revision of the sequencer logic to reduce the required amount of macrocells and interconnects². One modification made was to reduce the maximum length, i.e., the number of resolution units of Bitmaps. Because the control system is aimed at high-speed applications, Bitmaps longer than 1024, which give 12.8 μ s, are not expected. The logic block included in the original sequencer logic to control the retrieval of the image frames stored in static RAMs was also removed. It is the responsibility of the Image RAM Addresser described in sub-section 4.7.4. In addition, the control of the integration time was handed over to the interface card as described previously.

After the successful fitting of the sequencer logic, it was simulated with artificial inputs. The simulation result revealed that inaccurate timing between macrocells assigned to the Bitmap counter in the BM_CNT function block caused the unexpected behaviour of its outputs. This timing problem was resolved by enforcing the allocation of macrocells assigned to the Bitmap counter into adjacent LABs, thus minimising the

¹ The maximum internal array clock frequency is measured with a 16-bit loadable, enabled, up/down counter programmed into each LAB.

² The report file generated after successful fitting of the sequencer logic showed that only 3 % of available shareable expanders were left whereas 70 % of macrocells were used.

propagation delay between them that operate at the highest speed. The enforcement was accomplished with the text design file at the top level of the sequencer logic, which allocates specific macrocells to registers defined in the graphic design file.

4.4.2.2 Interface with the Host PC and DAC Control

The information required for the system hardware is transmitted from the interface card on the host PC via one of two 16-bit wide data buses. The data bus and the OBF signal from the interface card are coupled through nine opto-isolators. When the LSB of the data bus, D0 and the OBF signal are LOW, the data sets are interpreted by the INTFACE block. In the INTFACE block, four bits are decoded to generate various control and write ENABLEs. The remaining 11 bits are fed into other function blocks and they represent a DN. The OBF signal is reflected directly back to the interface card for the acknowledgement as an active LOW pulse.

When D1 and the OBF signal are LOW and D0 is HIGH, the lower byte is accepted by the DAC_CTRL block. By decoding three bits, it provides write ENABLEs for up to seven octal DACs and the control signal to turn them on and off. One of the write ENABLEs is assigned to the octal DAC on the Bias Board, two of them to two octal DACs on each Clock Board, and the remaining two are reserved. The remaining three bits represent the address bus to the octal DACs. The DAC_CTRL block also provides ENABLEs for octal buffers on Bias and Clock Boards. The upper byte, which represents the DN written in each DAC to define its output voltage level is transmitted directly from the Bitmap Generation Board to other boards.

4.4.2.3 Bitmap RAMs and Their Mapping

Three high-performance static RAMs with an organisation of $8\text{ K} \times 8$ bits are used to store 24-bit wide Bitmaps. The use of each Bitmap channel is described in sections 4.6 and 4.7 (or see Figure [6.8]). The sequencer controls their address bus and output ENABLEs and thus retrieves the Bitmaps from them in the correct sequential order for two Clock Boards and two Signal Processing Boards. They have the minimum readout cycle time of 12 ns that is smaller than the Bitmap resolution unit of 12.5 ns.

The lower half of each Bitmap RAM is divided into four equal-sized blocks to store byte wide streams of frame-transfer, vertical transfer, horizontal transfer, and purge

Bitmaps. Because the sequencer defines the number of resolution units of Bitmaps with 10 bits, each block is limited to have the maximum storage space of 1024 bytes. In writing and readout procedures of the sequencer, RD_CTRL and WR_CTRL function blocks ensure the access to the correct storage block by manipulating two MSBs of the 12-bit wide address bus. Other bits are provided by the Bitmap counter in the BM_CNT function block. The schematic diagram in Figure [4.5] shows the organisation and mapping of each Bitmap RAM.

During the testing of the Bitmap Generation Board, some considerable ringing and over and under-shoots were observed on the address bus between the sequencer and Bitmap RAMs. In order to tackle this problem, a clamp circuit made of high-speed Schottky diodes was employed to restrict HIGH and LOW voltage levels of the master clock output. Various termination methods on the address bus were also tried. The termination method of matching source and load impedances with a serial resistor was found to be simple but effective. Its optimum value of 150 Ω was chosen after trying different resistances and observing ringing and over- and under-shoots.

4.4.2.4 Writing of System Parameters and Bitmaps

To generate Bitmaps in the correct sequential order, the sequencer needs the knowledge of the CCD physical format prior to the readout. It also needs the numbers of resolution units of Bitmaps or their lengths to manipulate the address bus of the Bitmap RAMs. In the writing procedure, various parameters related to the CCD format and Bitmaps are stored in the sequencer in addition to actual Bitmaps in the Bitmap RAMs. These parameters are the number of repetitions of frame-transfer Bitmaps, the numbers of vertical and horizontal transfers, and the numbers of resolution units of all Bitmaps except for the purge Bitmaps. They are loaded into appropriate binary counters to generate Bitmaps during the readout procedure. Whereas the length of Bitmaps is limited to 1024 as described previously, other parameters can be as large as 2048 (11 bits). This feature was implemented to grant the control system the capability to read out other high-speed CCDs of relatively large formats. Seven write ENABLEs are provided by the INTFACE block to store them in the appropriate latches of various function blocks. The number of repetitions of frame-transfer Bitmaps are stored in the latches of the FT_CNT block whereas the numbers of

vertical and horizontal transfers in those of LT_CNT and HT_CNT blocks, respectively. The latches in the BM_CNT block store the lengths of Bitmaps.

Once storing these parameters in latches is completed, actual Bitmaps can be written into any one of three Bitmap RAMs. This process requires the EN_WR signal to be initiated and a Bitmap RAM to be selected. The Bitmap counter in the BM_CNT function block is loaded with the numbers of resolution units of Bitmaps in the order of frame-transfer, vertical transfer, horizontal transfer, purge Bitmaps. The WR_CTRL block provides SEL_HI and SEL_LO signals for the manipulation of two MSBs, M11 and M10, on the address bus to ensure the access to Bitmap storage blocks in this order. Other bits are supplied by the Bitmap counter that is counted down at the falling edge of the OBF signal from the interface card. The RAM_CTRL block generates write ENABLEs for the Bitmap RAMs with the OBF signal delayed by one period of the master clock to assure the proper timing between the sequencer and Bitmap RAMs.

4.4.2.5 Readout of the CCD

At the completion of writing system parameters, the readout of the CCD is initiated by activating the RD_EN control signal. The sequencer supports two readout modes; continuous and burst. The user must select the desired readout mode by defining the 7-bit DN stored in the latches of the RD_CTRL block. If its MSB is HIGH, the CCD is read out in the continuous mode regardless of other bits. Otherwise, the DN represented by the remaining 6 bits gives the number of image frames to be acquired in the burst mode. In both modes, the RD_END signal, which resets all counters and loads them with the appropriate DNs, is generated at the acquisition of each image frame. In the burst mode, the REAL_END signal is generated when the given number of image frames are acquired. To terminate the readout procedure in the continuous mode, the user must reset the sequencer by enabling the USER_RST signal.

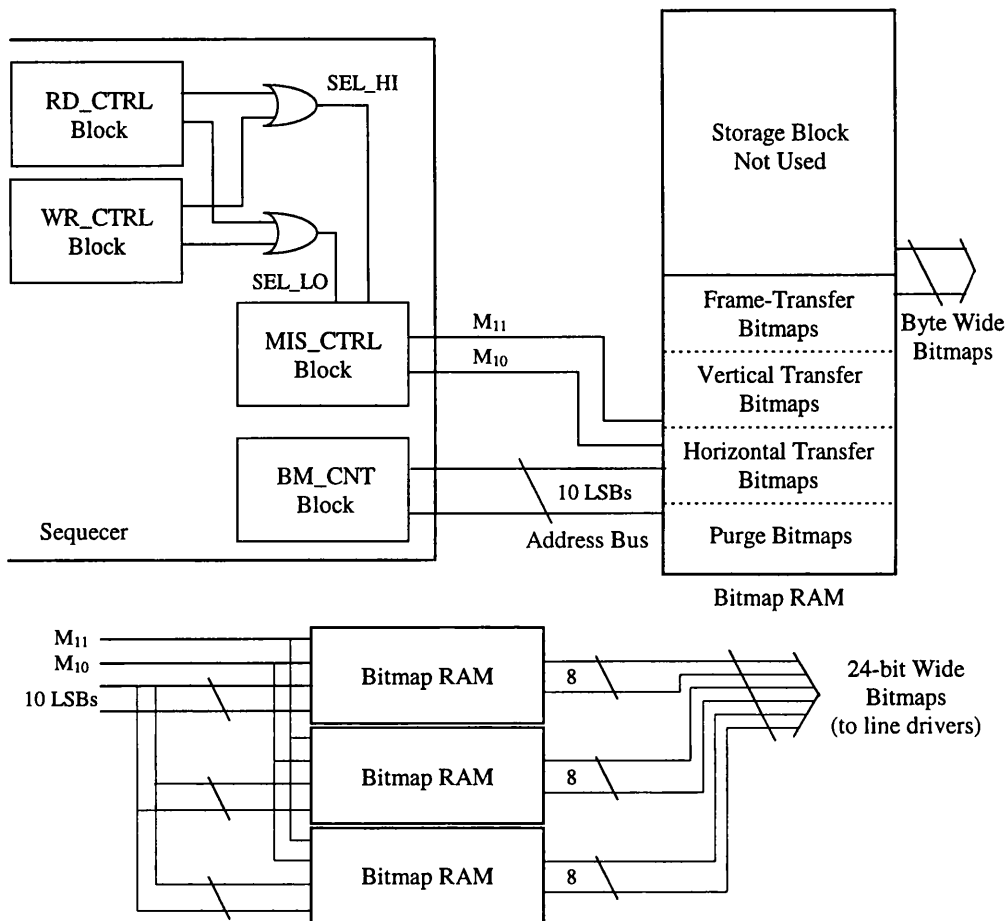


Figure [4.5] Organisation of each Bitmap RAM and its mapping by the sequencer

The acquisition of a single image frame begins with the frame transfer, which is accomplished by the repetition of frame-transfer Bitmaps for the number of times stored in the latches of the FT_CNT block. It is followed by sampling the signal charges with the repetition of vertical and horizontal transfer Bitmaps. In this procedure, the master clock drives the 10-bit Bitmap counter in the BM_CNT block. To load the correct number of Bitmap resolution units into this counter, the RD_CTRL block uses a combinational logic of carry-outs from three counters in FT_CNT, LT_CNT, and HT_CNT blocks. The MIS_CTRL block uses two signals, SEL_HI and SEL_LO, generated by the RD_CTRL block to manipulate M11 and M10 and thus to ensure the access to the correct storage block of Bitmap RAMs. All three Bitmap RAMs are read out simultaneously, which gives 24-bit wide streams of Bitmaps as shown in Figure [4.5].

Functionally, the carry-out from the Bitmap counter serves as the clock to two counters in FT_CNT and HT_CNT blocks because the frame and horizontal transfer Bitmaps are repeated continuously for the given numbers of times unlike the vertical transfer Bitmaps. The carry-out is also fed back to re-load the Bitmap counter with the appropriate number of Bitmap resolution units. The loaded number of resolution units is decided by carry-outs from three counters in FT_CNT, LT_CNT, and HT_CNT blocks. It was uncovered during the simulation, however, that the Bitmap counter was re-loaded before the carry-outs from three counters changed. Obviously, this phenomenon was distinctive at the beginning of each transfer. A simple look-ahead end-of-count-down logic was implemented from the outputs of the Bitmap counter to generate a pulse prior to its carry-out. The look-ahead pulse serves as the clock to two counters mentioned above, which ensures the re-loading of the Bitmap counter with correct DNs. However, it demands that the Bitmaps are longer than 3 because the look-ahead pulse is generated when the Bitmap counter reaches 2.

The sequencer supports the purging of the CCD ahead of the normal readout. This function is enabled by the EN_PURGE control signal. In case it is enabled, the sequencer addresses the Bitmap RAM block where the purge Bitmaps are stored for the acquisition of the first image frame instead of the block where the horizontal Bitmaps are stored. The flow chart in Figure [4.6] shows the CCD readout procedure with the purging disabled.

4.4.2.6 Miscellaneous Functions

All latches and counters in the sequencer can be explicitly reset by transmitting five LOW LSBs on the 16-bit wide data bus from the interface card and thus generating the USER_RST signal. The sequencer must be reset for its reliable operation and the safety of the CCD before any modification of system parameters is made or different Bitmaps are written into the Bitmap RAMs. The reset function is also used to terminate the readout procedure in the continuous mode.

As explained in section 4.2, the sequencer supports the function to manipulate a shutter mechanism when necessary. It supplies the active HIGH square pulse, SHUTTER that commences at the completion of the frame transfer. Its rising edge is used to trigger one of two timers on the interface card. The SHUTTER pulse goes LOW

at the end-of-integration signal, i.e., the carry-out generated by the timer. At the completion of the acquisition of each image frame, the acquisition of the next one is initiated by the AND-gated logic of the end-of-integration and RD_END signals. In case the integration time is longer than the time required to read out a image frame, the acquisition of the next one is initiated by the end-of-integration signal. The integration time is defined by the count-down DN stored in the integration timer. If an external shutter mechanism is unnecessary, the timer must be loaded with the DN to minimise the overhead on the readout time by producing the end-of-integration signal ahead of the RD_END signal. In addition, it must be noted that the timer fails to produce its carry-out when the count-down DN is too small, which causes the readout procedure to cease.

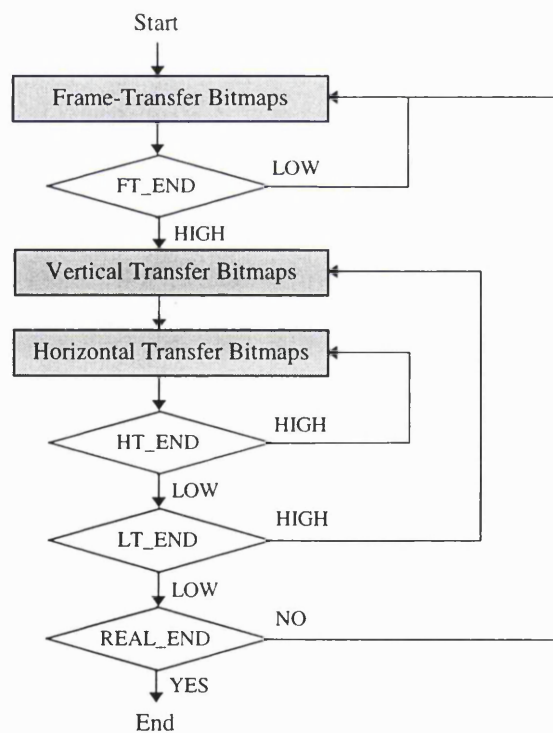


Figure [4.6] CCD readout procedure with the purging disabled

4.4.3 Other Components

The interface card in the host PC and the Bitmap Generation Board are coupled through ten bi-opto-isolators, HP2630. They are employed to electrically isolate the host PC and the Bitmap Generation Board. Eight of them are assigned to the 16-bit wide data bus, one to the OBF and end-of-integration signals, and the last to the SHUTTER signal that is used to initiate the integration timer on the interface card. Three octal buffers interface the byte wide data bus from the sequencer to each Bitmap RAM. They prevent data contention between Bitmap RAMs in the writing procedure caused by the data bus sharing.

Due to its high-speed nature, the Bitmap Generation Board generates a substantial amount of high-frequency switching noise. To minimise the coupling of this switching noise to two Clock Boards and two Signal Processing Boards during the transmission of Bitmaps, differential line drivers and receivers are employed. Two quad CMOS differential line drivers, DS90C031 are assigned to each Bitmap RAM. They support the data transmission rate up to 155.5 Mbps (77.7 MHz) and generate ± 350 mV differential outputs from either TTL(transistor-transistor logic) or CMOS inputs. These differential outputs are transmitted to other boards over 100 Ω twisted pairs. The TTL-compatible Bitmaps are retrieved at the receiving end by matching differential line receivers.

Even after the implementation of serial resistors between the sequencer and Bitmap RAMs as described previously, ringing and over and under-shoots persisted between Bitmap RAM outputs and differential line driver inputs. Additional 82 Ω serial resistors whose optimum value was found in a similar manner to that described in subsection 4.4.2.3 were added between them.

4.5 Bias Board

4.5.1 Overview

To operate a CCD, constant biases are required for CCD nodes such as drain and source of the output transistor, drain of the reset transistor, and substrate. The Bias

Board is responsible for the provision of these constant biases. It can supply up to eight constant biases. Their voltage levels are defined by DACs that are controlled by the user through the control software without any modification of the system hardware. It consists of an octal DAC, an octal buffer, a voltage reference, a monostable multivibrator, and 16 op-amps as shown in Figure [4.7].

4.5.2 Octal DAC for CCD Biases

The octal DAC, AD7228KN is used to specify bias voltage levels of the CCD. Its address bus and write ENABLE are controlled by the Bitmap Generation Board. The upper byte of the 16-bit wide data bus from the interface card on the host PC is written into each DAC to define its output voltage level. With the +10 V reference voltage and +15 V power supply, each DAC can be programmed to produce a constant voltage level from 0 to +10 V with 8-bit resolution. Its output voltage is translated and amplified by op-amps to support bipolar CCD biases. The 3-bit wide address bits and the write ENABLE are transmitted through single wires and the DNs are interfaced by the octal buffer. The write ENABLE has the same LOW pulse width of 80 ns as the OBF signal originated from the interface card, which is shorter than the minimum width required by the octal DAC. The monostable multivibrator is employed to generate the ENABLE with longer LOW pulse widths.

At the initial design phase of the Bias and Clock Boards, a relay and its driver were employed to switch on and off the power supplies to op-amps by the DAC_ON/OFF signal from the sequencer. This feature was included to eliminate the possibility that the CCD could be damaged in the process of writing octal DACs. In this process, CCD clock gates can be applied with excessively high negative voltages prior to the application of the substrate voltage due to the incorrect writing order of octal DACs. However, during the testing of these boards, many octal DACs were found to be damaged before the power supplies to op-amps were switched on. It is believed to have been caused by op-amps that drew a large amount of currents from DACs beyond their current sourcing capability when their inputs were applied with their power supply pins floating.

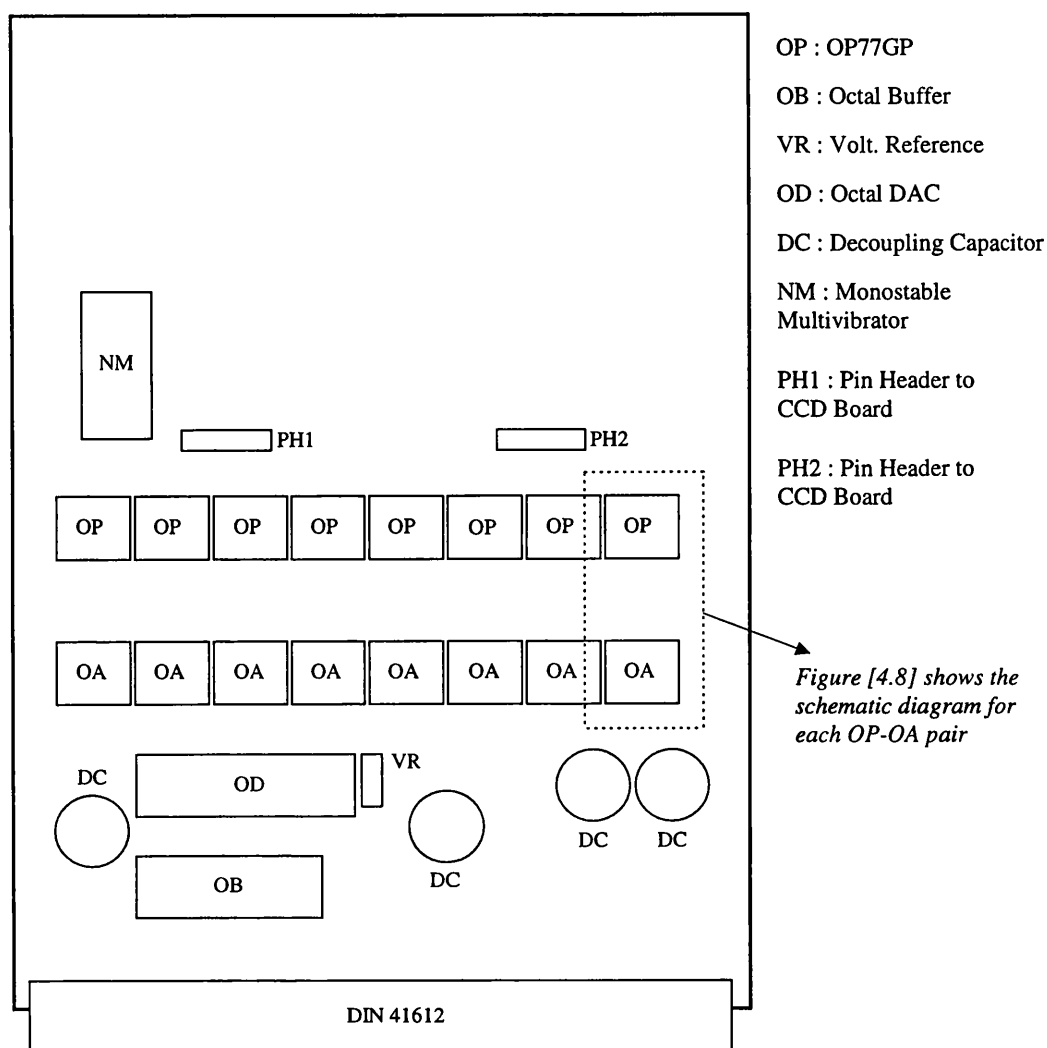


Figure [4.7] Layout of the Bias Board

4.5.3 Translation and Amplification of the DAC Output

The output from each DAC is translated and amplified by the two-stage op-amp circuit. To support the MPP operation of particular CCDs, the circuit was designed to be able to supply bipolar biases. The first stage translates and amplifies the DAC output voltage from the range of 0 to +10 V to that of -10 to +10 V. The second stage amplifies the output voltage to the range of -18 to +18 V or -20 to +20 V. The schematic diagram of the translation and amplification circuit is shown in Figure [4.8]. The small feedback capacitor, C, and the small serial resistor, R2, are included to isolate the highly capacitive CCD loads from the op-amp outputs. The upper and lower bias voltage levels are limited by the maximum rating of op-amp power

supplies at the second stage, which is ± 22 V. With a byte wide DN written into a single DAC and ± 18 V power supplies, the CCD bias voltage, $V_{\text{bias-out}}$, at the end of the second stage is approximately given by

$$V_{\text{bias-out}} = (\text{DN} - 127) \times 36/256.$$

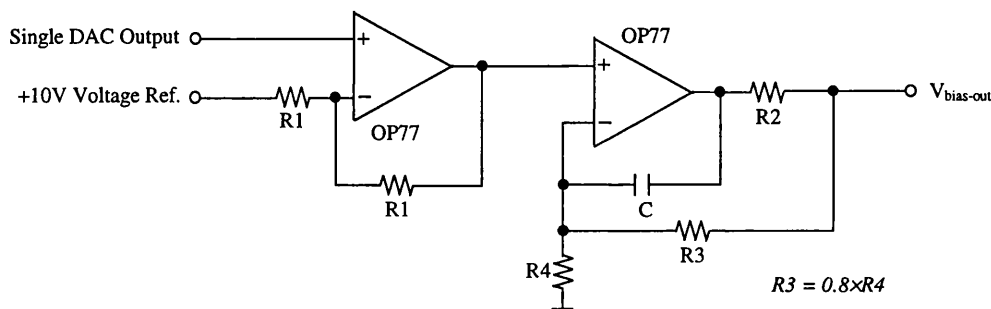


Figure [4.8] Diagram of the translation and amplification circuit to supply the CCD bias from the DAC output

4.6 Clock Boards

4.6.1 Overview

The sequential streams of Bitmaps generated by the Bitmap Generation Board give the precise timing pulse streams to readout a CCD. However, they must be converted to have the appropriate voltage levels to drive the actual CCD clock gates. It is the function of two Clock Boards to provide the clock pulse streams to the CCD. There are two Clock Boards in the system hardware, which are identical except for the clock driving circuit. Out of 24-bit wide Bitmaps or 24 Bitmap channels, each Clock Board accepts byte wide Bitmaps or 8 channels. The remaining byte wide Bitmaps are halved and transmitted to two Signal Processing Boards for control signals as described in section 4.7. Physically, one of two Clock Boards is dedicated to the vertical clock gates and the other to the horizontal ones. They are designed to incorporate the static DAC control method explained in Chapter 3. Each board can provide the clock pulse streams for up to 8 CCD clock gates, voltage levels of which are independently defined by two octal DACs. Each board consists of two octal DACs, an octal buffer, a

monostable multivibrator, two differential line receivers, four quad op-amps, eight high-speed MOSFET drivers, and several transistors. The layout of the Clock Board dedicated to the vertical transfer clock gates is shown in Figure [4.9].

4.6.2 Octal DACs with the Static DAC Control Method

On each Clock Board, there are two octal DACs, the control of which is accomplished in the same manner as the one on the Bias Board. Their byte wide DNs are interfaced by the octal buffer and write ENABLEs are supplied by the monostable multivibrator. One of them, HIGH DAC, is used to define the HIGH clock voltage levels of eight clock gates and the other, LOW DAC, to define their LOW clock voltage levels.

Eight outputs from the HIGH DAC can be programmed to have the voltage range of 0 to +10 V with the reference voltage of 10 V. Each output is buffered by a voltage follower that drives the base of a high current NPN transistor, ZTX851. Similarly, eight outputs from the LOW DAC change their polarity to have the voltage range of 0 to -10 V through inverters. Each op-amp inverter drives the base of a high current PNP transistor, ZTX951. The emitters of high current transistors perform the role of power supplies to the high-speed MOSFET drivers as shown in Figure [4.11]. They have the capability to source and sink high current for the MOSFET drivers, which is essential to drive the capacitive CCD clock gates. The ZTX851 can source current up to 5 A and the ZTX951 can sink up to 5 A.

Because the MOSFET driver switches between its positive and negative supplies to generate pulse streams, octal DACs eventually define the HIGH and LOW voltage levels of the CCD clock pulses and reflect the static DAC control method faithfully. For the given DNs written into two single DACs, HIGH and LOW levels of CCD clock pulses, $V_{\text{clock-high}}$ and $V_{\text{clock-low}}$, are approximately given by

$$V_{\text{clock-high}} = \text{DN} \times 10 / 256, \quad V_{\text{clock-low}} = -\text{DN} \times 10 / 256.$$

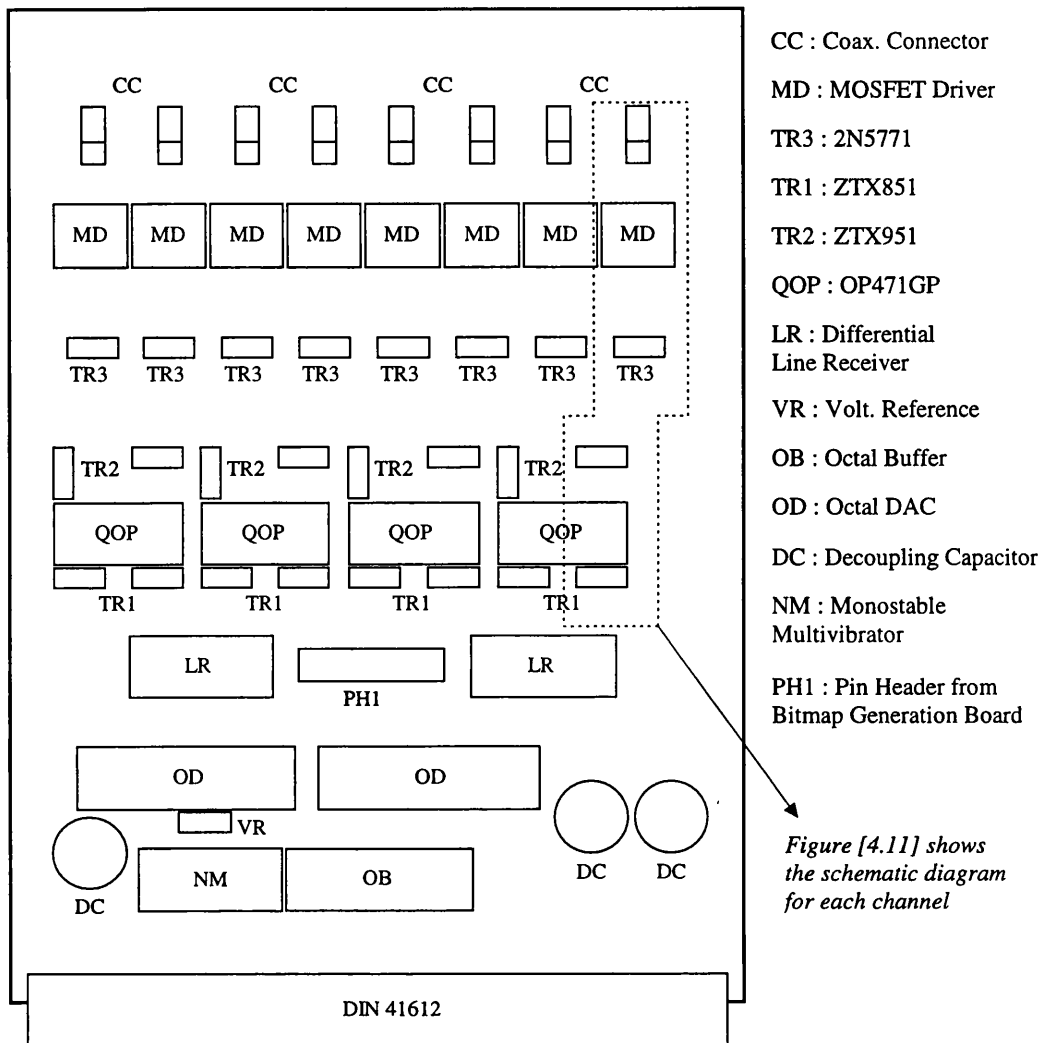


Figure [4.11] shows the schematic diagram for each channel

Figure [4.9] Layout of the Clock Board assigned to the vertical transfer clock gates

4.6.3 Generation of CCD Clock Pulse Streams

On each Clock Board, the differential streams of Bitmaps transmitted from the Bitmap Generation Board via twisted pairs are converted to TTL pulse streams by two quad CMOS differential line receivers, DS90C032. They are matched pairs of the differential line driver, DS90C031. For the satisfactory high-speed readout of a CCD, the clock driver must have the capability to source and sink high currents because the clock gates are intrinsically capacitive and switched at high speeds with relatively high voltage swings. To realize the frame rate of 1000 Hz for 100×100 pixels, the pixel sampling frequency must be about 10 MHz in the worst case of a single output node.

With the Bitmap resolution unit of 12.5 ns, it gives the length of the horizontal transfer Bitmaps of 8 (100 ns). Figure [4.10] shows a typical horizontal transfer clock pulse stream for a 3-phase CCD. It can be seen that at least 7 resolution units are needed to define it when the clock driver charges and discharges the clock gates within a single resolution unit.

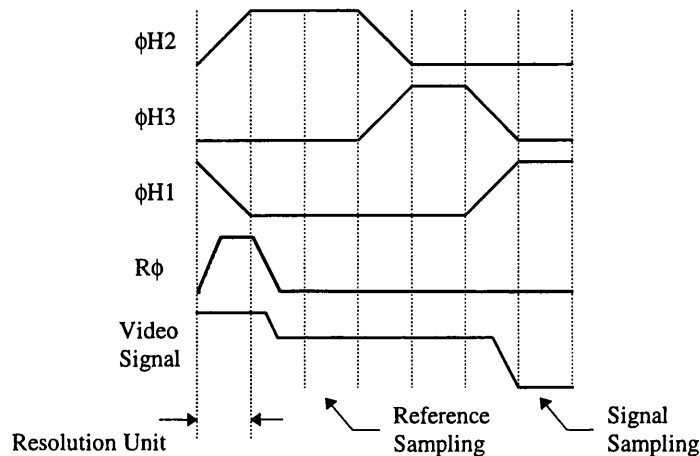


Figure [4.10] A typical horizontal transfer clock pulse stream for a 3-phase CCD. The pulse overlapping is 50 %. It is assumed that the clock gates are charged and discharged within 12.5 ns and the reset gate capacitance is much smaller than those of others. For reference and signal sampling, see section 4.7.

Various clock driving circuits were tried with discrete devices. However, due to the stray capacitance associated with the prototype module board, they failed to exhibit their specified performance and to meet the requirement in terms of the rise and fall times at the specified maximum voltage swing. Instead, the high-speed MOSFET driver, EL7212 was tried as the clock driver. It can handle a peak current up to ± 2 A and shows typical rise and fall times less than 10 ns with a 500 pF load. This characteristic was considered ideal for the control system taking the Bitmap resolution unit of 12.5 ns into consideration. However, it caused damage to many neighbouring devices when the LOW clock voltage is negative. The more detailed description and analysis on the clock driver built with the EL7212 is given in Chapter 6.

A new MOSFET driver, DS0026 was chosen to replace the EL7212 and the driving circuit was modified accordingly. It has specifications slightly inferior to the EL7212. It gives rise and fall times of 20 ns over a 1000 pF load and a peak output current of ± 1.5 A. Its maximum switching frequency, that eventually defines the maximum pixel sampling frequency, is dependent upon the power dissipation. Taking into account much smaller capacitances of the horizontal clock gates for high-speed CCDs than 1000 pF, the DS0026 was expected to meet the pixel sampling frequency of 10 MHz. It was verified experimentally as described in Chapter 6. The DS0026 also produces output pulses whose voltage levels are defined by its positive and negative power supplies.

An external capacitor or an equivalent circuit(i.e. AC-coupled) is needed to drive the DS0026 from TTL input pulses. DS0026's driven through external capacitors, however, require input TTL pulses running over a specific frequency to prevent the undesired discharge of capacitors. The minimum frequency of the driving input pulses is governed by the external capacitance. This characteristic is disadvantageous to drive the vertical clock gates because of the relatively long readout time associated with the horizontal transfers. Alternatively, a simple input circuit with a high-speed switching PNP transistor was tried. Its long turn-off time, however, revealed that it is not suitable for the horizontal clock gates.

In consequence, it was decided to dedicate one of two Clock Boards to the generation of the horizontal clock pulses and the other to that of the vertical ones. Each of the boards can handle up to eight Bitmap channels with DS0026's driven either by external capacitors or by transistor circuits. The external capacitors were chosen in view of the balanced rise and fall times and the minimum switching frequency required to prevent their discharge. It was necessary, however, to manipulate the horizontal transfer Bitmaps to prevent their discharge over the vertical transfer period. Figure [4.11] is the circuit diagram of a single clock driver for vertical clock gates. For the horizontal clock gates, the resistor, R1, and transistor, 2N5771, in the dotted box is replaced by the external capacitor. Eight CCD clock pulse streams from each Clock Board are transmitted to the CCD Board via single-ended miniature coaxial cables that have a characteristic impedance of 50 Ω .

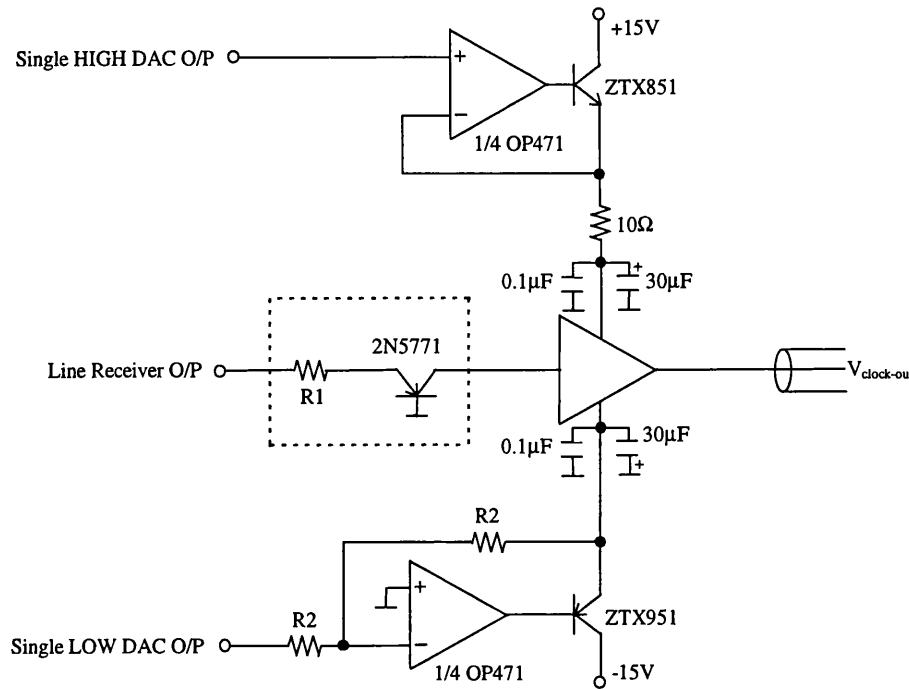


Figure [4.11] Circuit diagram of the vertical CCD clock driver

4.7 Signal Processing Boards

4.7.1 Overview

One of the advantages of the CCD is that it is an analog light-sensitive device that is controlled in a digital fashion. In the readout procedure of a CCD, the signal charges generated by incident photons and accumulated in potential wells over the integration time are transferred along pixels and sampled at its output node(s) sequentially. The sensing capacitor and the reset transistor at the output node sample the signal charges pixel by pixel and convert them into the video signal superimposed on the constant reset drain bias. The amount of the signal charges in a pixel is represented by the offset voltage from the constant reset drain bias. The video signal is generally buffered by one or two common-source on-chip amplifiers that have a low output impedance and wide bandwidth. Some CCDs employ non-destructive ‘skipper’ output amplifiers to reduce the noise from the amplifiers.

The offset voltage level can be retrieved from the output video signal to yield the information on the amount of the signal charges and eventually the number of photons accumulated in pixels. The retrieval is the main function of two Signal Processing Boards. DNs produced from sampling the video signals are stored in four static RAMs named the Image RAMs on two Signal Processing Boards. Each of them can handle two CCD output signals independently in parallel. One of them, the Signal Processing Board #1, has an EPLD (Image RAM Addresser) programmed with the logic to control the storage and retrieval of the sampled DNs. The other, the Signal Processing Board #2, has four multiplexers to transmit the sampled DNs (or more simply pixel values) retrieved from the Image RAMs to the interface card on the host PC over the byte wide data bus.

4.7.2 Video Signal Processing

On two Signal Processing Boards, all CCD output video signals are processed and sampled in the same manner by the analog circuit shown in Figure [4.12]. It consists of two high-speed op-amps, two high-speed S/H(sample-and-hold) amplifiers, a single DAC, and an ADC(analog-to-digital converter). The video signal is transmitted from the CCD board via a miniature coaxial cable. The circuit was designed to process and sample video signals that are AC-coupled by the pre-amplifiers.

The non-inverting amplifier built with the high-speed op-amp, AD811, amplifies the input video signal by a factor of $(1+R_2/R_1)$. Because the AD811 is a current feedback op-amp, its feedback resistor, R_2 defines the bandwidth of the non-inverting amplifier. The selection of a proper feedback resistance, therefore, can contribute to the reduction of the high frequency noise. The amplifier at the first stage of the signal processing chain is normally designed to have a bandwidth with the 3 dB cut-off frequency three times higher than the maximum pixel sampling frequency.

Following the amplifier, a simple CDS(correlated double sampling) circuit eliminates the reset noise originated from the process of resetting the sensing capacitor. The amplified video signal is sampled twice by two S/H amplifiers, SHC605. With its versatile architecture, the SHC605 can be readily configured to have non-inverting, inverting, or differential gain. In the high-speed CCD control system, it is configured

as a simple S/H amplifier with unity gain. The amplified video signal is sampled by one of two S/H amplifiers just before the signal charges are dumped into the sensing capacitor: the reference sampling. It is sampled again by the other just before the sensing capacitor is reset by the reset transistor: the signal sampling. Subtracting the output of one of two S/H amplifiers from that of the other eliminates the reset noise and produces the offset voltage proportional to the amount of signal charges accumulated in pixels.

The reference sampling is performed by the top S/H amplifier in Figure [4.12] and the subtraction is accomplished by the following op-amp. It also adds the negative DC offset specified by the single DAC (video offset DAC) to the subtracted video signal. The negative DC offset has the range of 0 to -2.56 V. It is added to the video signal to prevent the ADC input signal from going positive and thus protects the ADC. The control of four video offset DACs is performed by the Image RAM Addresser as described in sub-section 4.7.4. Processed by the CDS circuit, the video signal is digitised by the high-speed ADC, AD9048. The -2 V reference to the ADC is supplied by a stable circuit constructed with an op-amp, LM741, and a transistor, 2N3901. DNs produced by four ADCs are stored in four Image RAMs on two Signal Processing Boards.

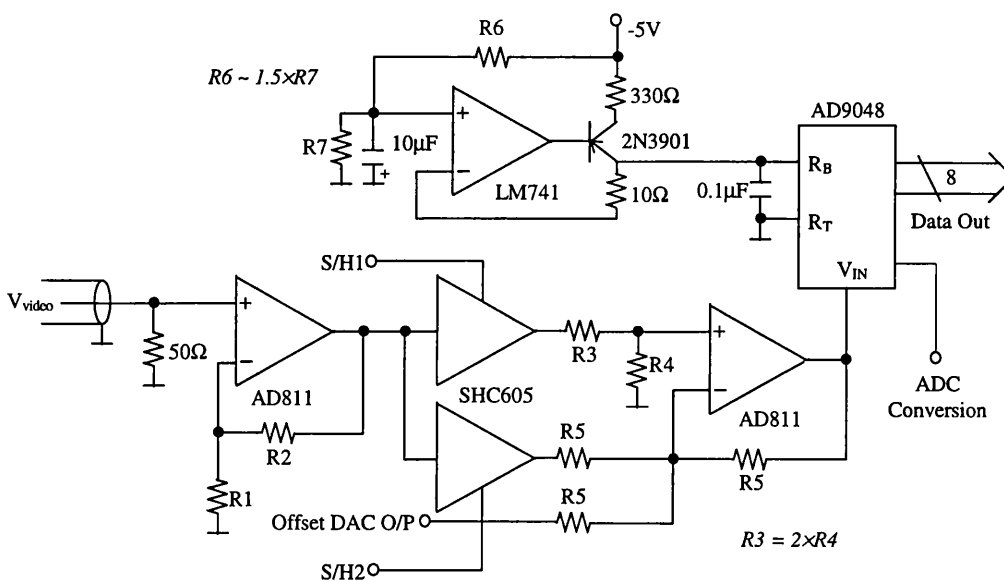


Figure [4.12] Circuit diagram to process and sample the CCD output video signal

4.7.3 Control of Video Signal Sampling and ADC Conversion

One of the advantages of the Bitmap control method is that the timing pulses for control signals can be included in Bitmaps. The sequencer generates 24-bit wide Bitmaps out of three Bitmap RAMs. 16-bit wide streams of them are allocated to define the CCD clock timing pulses on two Clock Boards. The remaining byte wide streams are used to define the timing pulses of various control signals. As can be seen in Figure [4.12], the circuit for the signal processing and sampling requires three control signals; two S/H signals and one ADC conversion signal. In addition, the Image RAMs require write ENABLEs to store the digitised pixel values.

Each half of byte wide Bitmaps is assigned to each Signal Processing Board to define three control signals. They are transmitted from the Bitmap Generation Board via twisted pairs. The quad differential line driver on each board retrieves the TTL timing pulses of control signals. Because of the high-speed nature of the control system, the horizontal transfer Bitmaps normally have short lengths. Therefore, the time available for the digitisation of video signals is often shorter than the definite ADC conversion time. To ensure the complete digitisation, the ADC conversion signal from the horizontal transfer Bitmaps is applied to ADCs after some time delay.

The delay line employed on each board has several output taps and the ADC conversion signal is taken from one of them to give 30 ns delay. It effectively eliminates the need to allocate some resolution units to the horizontal transfer Bitmaps for the ADC conversion. The last output tap of the delay line also provides write ENABLEs to the Image RAMs because the ADC does not generate the end-of-conversion signal. The remaining Bitmap channel on the Signal Processing Board #1 is used by the Image RAM Addresser to control the Image RAM address bus in storing the pixel values as described in the following sub-section. Refer to Figures [4.6] and [6.8.b] for the timing of two sampling and ADC conversion signals.

4.7.4 Image RAM Addresser, EPM5128

To store the pixel values sampled and digitised in the CCD readout procedure, a static RAM (Image RAM) organised as $32\text{ K} \times 8$ bits is assigned to each ADC. The Image

RAM Addresser on the Signal Processing Board #1 is an EPLD, EPM5128 that is programmed with the logic to perform the manipulation of the address bus shared among four Image RAMs. It controls the address bus both in the CCD readout procedure and in the process of the retrieval of the stored pixel data. It also controls four video offset DACs and four multiplexers on the Signal Processing Board #2.

Initially, it was tried to integrate its logic functions into the sequencer. However, it was unable to secure the required amount of logic cells. It was also considered impractical because of the physical separation of the Bitmap Generation Board and the Signal Processing Board #1 and the large number of possible interconnects between them. The EPM5128 is a member of the MAX 5000 EPLD family manufactured by Altera. It has 128 macrocells combined into eight LABs. The logic of the Image RAM Addresser was simulated and programmed into the EPM5128 with the development package, MAX+PLUS. The layout of the Signal Processing Board #1 where the Image RAM Addresser resides is shown in Figure [4.13].

4.7.4.1 Logic Overview

The Image RAM Addresser consists of several function blocks. Unlike the sequencer, it is not driven by a dedicated master clock. To perform its logic functions, it uses one channel out of byte wide Bitmaps assigned to the Signal Processing Board #1 and the OBF signal from the interface card on the host PC. Each function block is briefly described below and its schematic block diagram at the top level is given in Appendix A together with its input and output pin assignment.

- **INTERFACE** block decodes 7 LSBs out of the 15-bit wide data bus transmitted from the Bitmap Generation Board and generates the control and write ENABLEs for other function blocks. It also interfaces DNs represented by the remaining 8 bits to them.
- **CLOCK** block generates the clock to the binary counter in the ADDRESS block in both processes of storing the pixel values in the Image RAMs and of retrieving them for the transmission to the host PC.
- **RD_CTRL** block controls the sequential readout process of four Image RAMs by providing output ENABLEs and manipulates two input selection signals,

MUX_A and MUX_B of four multiplexers on the Signal Processing Board #2. They are used to select the Image RAM to be read out.

- ADDRESS block has latches to store the DN that represents the top address defined by the maximum number of image frames to be acquired and the CCD format. Its binary counter is loaded with the DN at the initiation of processes to store and retrieve the pixel values. Being counted down at the completion of storing and retrieving each pixel value, its outputs constitute the 16-bit wide address bus to the Image RAMs.

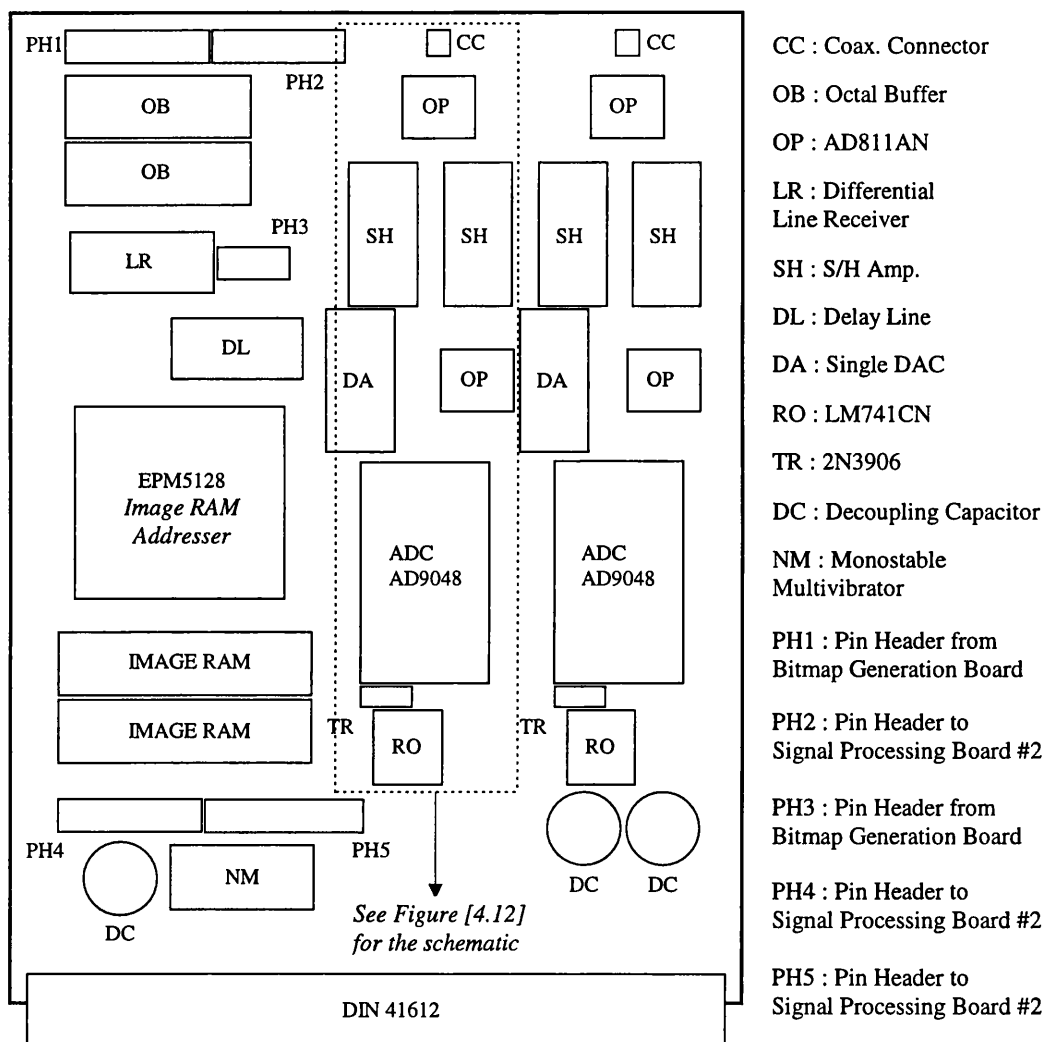


Figure [4.13] Layout of the Signal Processing Board #1

4.7.4.2 Interface with the Host PC and Offset DAC Control

The Image RAM Addresser uses 7 LSBs and 8 MSBs out of the 16-bit wide data bus and the OBF signal transmitted from the interface card. They are interfaced by two octal buffers on the Signal Processing Board #1. 7 LSBs are interpreted by the INTFACE block to generate the control and write ENABLEs for other blocks and four video offset DACs. When two LSBs, D0 and D1 are HIGH and the OBF signal, named WR_STBE, is LOW, the remaining 4 bits are decoded. 8 MSBs are either relayed to the ADDRESS block to define the 'top address' of the Image RAMs or used to define the output voltage levels of four video offset DACs, AD558's. Like octal DACs in other module boards, their write ENABLEs are made to have longer LOW pulse widths by monostable multivibrators.

The upper half of the other 16-bit wide data bus provided by the interface card is configured to receive the pixel data from the Image RAMs. The transmission is performed in the strobe mode and the STB signal, named RD_STBE, is reflected on the Signal Processing Board #1 to provide the active LOW IBF signal to the interface card.

4.7.4.3 Operation of the Image RAM Addresser

The amount of storage space available in the Image RAMs limits the number of image frames they can store. It is also a function of the CCD format (the numbers of vertical and horizontal transfers) and the ADC sampling resolution. All of these parameters are either determined during the design of the system hardware or given to the sequencer by the user. The system hardware and the 64×64 CCD, for instance, give a maximum number of seven image frames when they are stored frame by frame without any gap. The control software calculates the maximum number of sampled pixels that can be stored in the Image RAMs from the given parameters. It transmits the DN that represents this number to the Image RAM Addresser ahead of the CCD readout procedure.

The Image RAM Addresser uses this 16-bit wide DN to control the address bus to four Image RAMs. It is transmitted from the interface card in two bytes and stored in the latches of the ADDRESS block. At the initiation of the CCD readout procedure, it is loaded in the 16-bit binary counter, which is counted down at the sampling of each pixel. Because the ADC has the digitisation resolution of 8 bits, the outputs from the

counter are directly mapped onto the address bus to Image RAMs. In other words, the sampled pixel values are stored from the 'top' to the 'bottom' of the Image RAMs, which is why the DN is called the 'top address'.

One channel out of Bitmaps is used as the clock pulse to count down the 16-bit counter during the horizontal transfers. Clock pulses must have the HIGH pulse width sufficiently long to satisfy the setup and hold times of the EPM5128. The propagation delay in the EPM5128 and the Image RAMs also demands the interval allocated between the clock pulse and the S/H control signals to be sufficiently long. As can be easily deduced, old image frames are overwritten by new ones when more image frames are read out than the maximum number that the Image RAMs can store.

In its readout process, the Image RAM Addresser retrieves the image frames from the Image RAMs and transmits them to the interface card pixel by pixel. Because it is not required to read the pixel data out of them at high speed, their outputs are multiplexed and transmitted via the upper half of the 16-bit wide data bus of the interface card. Four 2-bit 4-to-1 multiplexers are employed on the Signal Processing Board #2 to multiplex four byte wide output buses to the byte wide data bus. Four Image RAMs are read out one by one in any order by the control software. According to the Image RAM selected by the user, the RD_CTRL block manipulates two input selection signals of multiplexers, MUX_A and MUX_B. The RD_CTRL block also provides the output ENABLEs to the selected Image RAM. The pixel values in the selected Image RAM are retrieved in a similar manner to the way they are stored. In this process, instead of one of the Bitmap channels, the STB signal from the interface card drives the 16-bit down counter in the ADDRESS block. The address bits for the Image RAMs are transmitted from the Signal Processing Board #1 to #2. The output data from all Image RAMs are multiplexed on the Signal Processing Board #2 whose layout is shown in Figure [4.14].

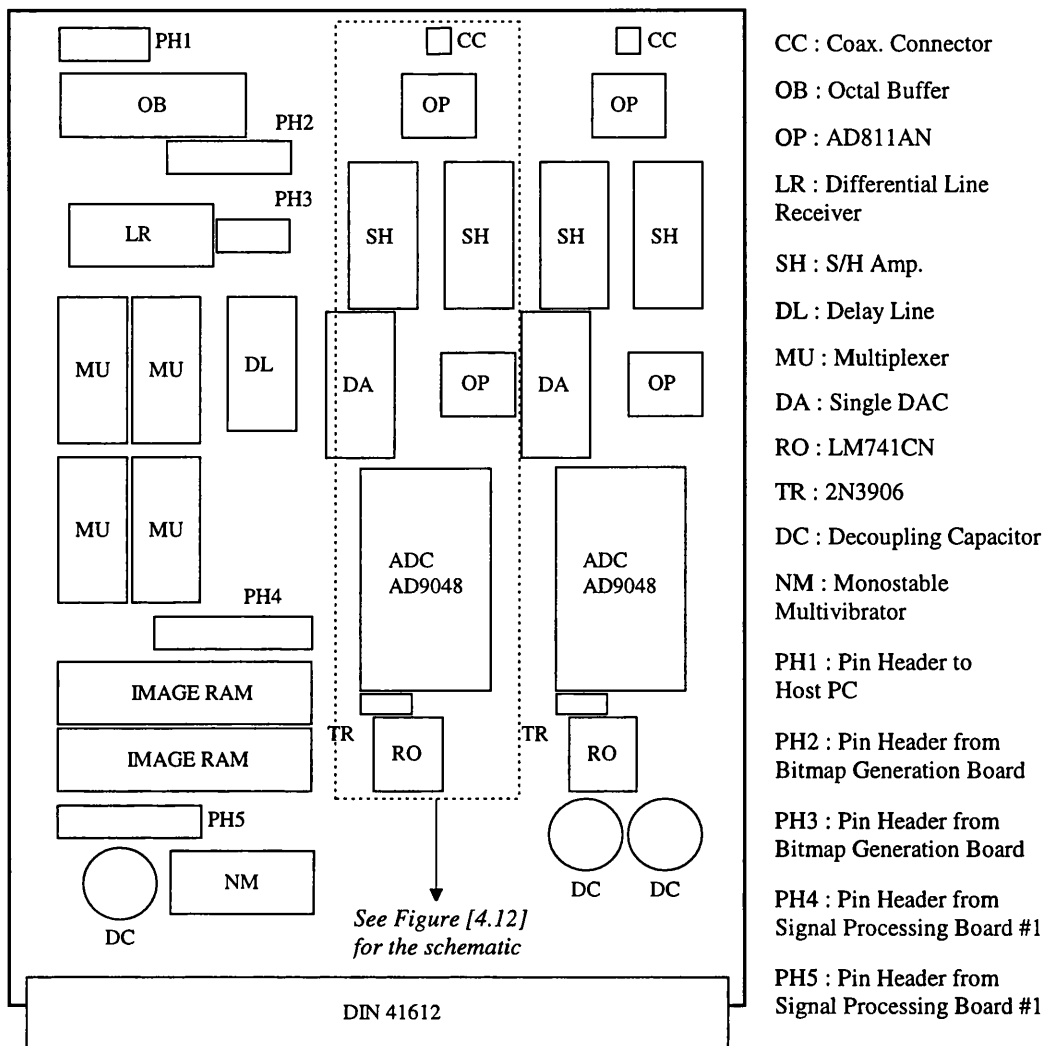


Figure [4.14] Layout of the Signal Processing Board #2

Graphical User-Interface and Control Software

The control system incorporates the Bitmap and static DAC control methods and thus allows the user to manipulate the CCD clock and control timing pulses and the clock and bias voltage levels. The sequencer also has the capability to drive high-speed CCDs of different physical formats. Because the system parameters provided by the control system are controlled without any modification of its hardware, an easy-to-use and well-structured control software environment is essential to make use of its versatility to the highest degree. The graphical user-interface(GUI) developed in this work reflects the design ideas embodied in the system hardware. It gives a user-interface environment that presents the abstraction of the system parameters with graphical objects. The user can easily manipulate any of them independently with the mouse pointing device and the keyboard. It also includes the Bitmap Editor that was developed to support the generation and modification of Bitmaps in a graphical style. The control software routines for the system hardware are included in the GUI, which provides an integrated control environment to the user.

The need for the graphical Bitmap Editor was brought forward from the author's past experience with the low-noise mosaic CCD control system. In that system, the user must define various Bitmaps as streams of characters within the source codes of the control software. Therefore, in the process of evaluation and optimisation, the source codes must be modified and re-compiled frequently. In addition, it is highly susceptible to human errors because the Bitmaps as streams of characters cannot give

the user the solid visualisation of the actual timing pulses. With the Bitmap Editor, the user can define, examine, and modify the Bitmaps represented graphically by streams of pulses. It ensures the more reliable and predictable operation of the control system. However, the solid understanding of the system hardware is needed to maximise the usefulness of the Bitmap Editor. Figure [5.1] illustrates the components of the GUI and control software and their interaction with the user and the system hardware.

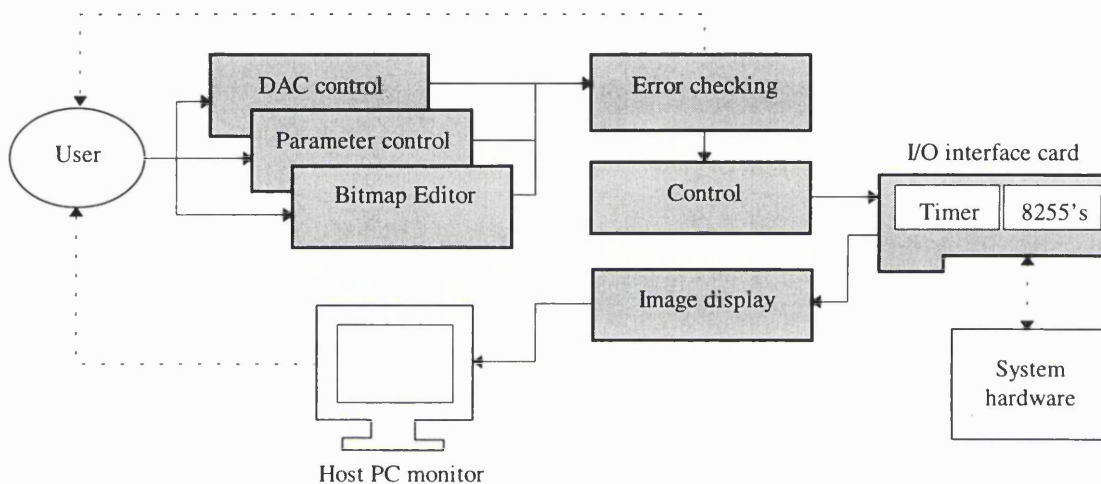


Figure [5.1] Components of the GUI and control software and their interaction with the user and the system hardware

At the time of the development of the GUI, there were many programming languages such as the Visual C++ that provide convenient and powerful environments to a programmer who intend to develop a GUI based on the Microsoft Windows operating system. In terms of the appearance of the end product and the time and effort needed for source-coding, they seemed attractive. However, it was believed that the attainment of the knowledge and experience in programming is more significant. Rather than being source-coded based on functions or objects provided by the programming language, the GUI was source-coded from the lowest level possible with the object-oriented C++ programming language. It inherently imposes the modular, structured, and 'top-down' source codes, which coincides with the design philosophy of the control system. The

compiler used is the version 3.1 of the Borland C++ programming language running on the host PC.

In section 5.1, a brief description on various classes for the GUI and control software is given following an introduction to the C++ programming language. Because they are building blocks of the GUI and control software, the understanding of their structure and functionality is prerequisite. The Bitmap Editor and the main control routines are described in sections 5.2 and 5.3, respectively.

5.1 Classes for GUI and Control Software

The C++ programming language was developed in the early 80s. It is an evolution of the C programming language that extends it in three important ways; support of abstract data types, support for object-oriented design and programming, and various improvements over the existing C[66]. In addition to the pre-defined set of data types, C++ supports the class mechanism that allows the programmer to define new abstract data types. An abstract data type is defined as a collection of data and operations on that data. A class is an aggregate of named data elements (data members) and a set of operations designed to manipulate that data (member functions). The developer of a class can control the access to its data members and member functions, which is referred to as *information hiding*.

Object-oriented programming is characterised by *inheritance* and *dynamic binding*. The inheritance mechanism extends abstract data types to allow for type/sub-type relationships. Through the inheritance, a class can inherit the selected data members and member functions of other classes. In C++, the inheritance is implemented through the mechanism of class derivation. If a class inherits the members of other classes, it is derived from them and they are its base classes. The other characteristic, dynamic binding transfers the task of resolving data types from the programmer to the compiler, and thus encapsulates the implementation details within an inheritance hierarchy. In C++, it is supported by the mechanism of virtual member functions. They are special class member functions that define type dependent operations in an inheritance hierarchy. For example, when several classes inherit a virtual member

function from a base class, the invocation of the virtual function through a pointer to the base class is resolved at run time to the appropriate derived class.

5.1.1 Overview

Various classes were defined for the GUI and control software of the high-speed CCD control system. They are categorised into four groups as listed below except for the IconList class. Figure [5.2] is the inheritance tree of the classes. In this figure, dotted arrows indicate classes used as data members by others. The description of each class is given in the following sub-sections. The GUI and control software operates with objects of these classes. An object of a specific class is normally referred to as an *instantiation* of the class. In other words, an object of a class is *instantiated* when it is created. Appendix B gives the header files in which the classes are declared.

- Classes to perform graphical functions
Graphics, Text
- Classes to handle user activities
Events, Mouse
- Classes to represent static or dynamic information
Icon, Button, InputBox, PWinExitBut, UInputBox, FInputBox
- Classes to perform tasks
MenuBar, CommBox, OKBox, YNBox, StrBox, PopupWin, PFileWin, SetupWin, MainWin, BMEWin, DACSetupWin, ParaSetupWin

5.1.2 Classes for Graphical Functions

The Graphics class provides the fundamental graphical routines to objects of other classes by invoking the functions defined in the super-VGA graphics library. The graphics library gives the programmer a transparent access to twenty different super-VGA graphics cards. Its identification routines can identify not only the graphics card, but also the amount of video memory, the type of processor, and the installed pointing devices. It contains functions to draw all major graphical primitives, such as lines, boxes, and circles on the screen. In addition, it provides functions to perform the

modification of the screen palette, writing texts on the screen with a choice of fonts, and the modification of the reaction of primitives. Finally, the support for two-dimensional and three-dimensional coordinate systems is provided including rotation, translation, scaling, and projection.

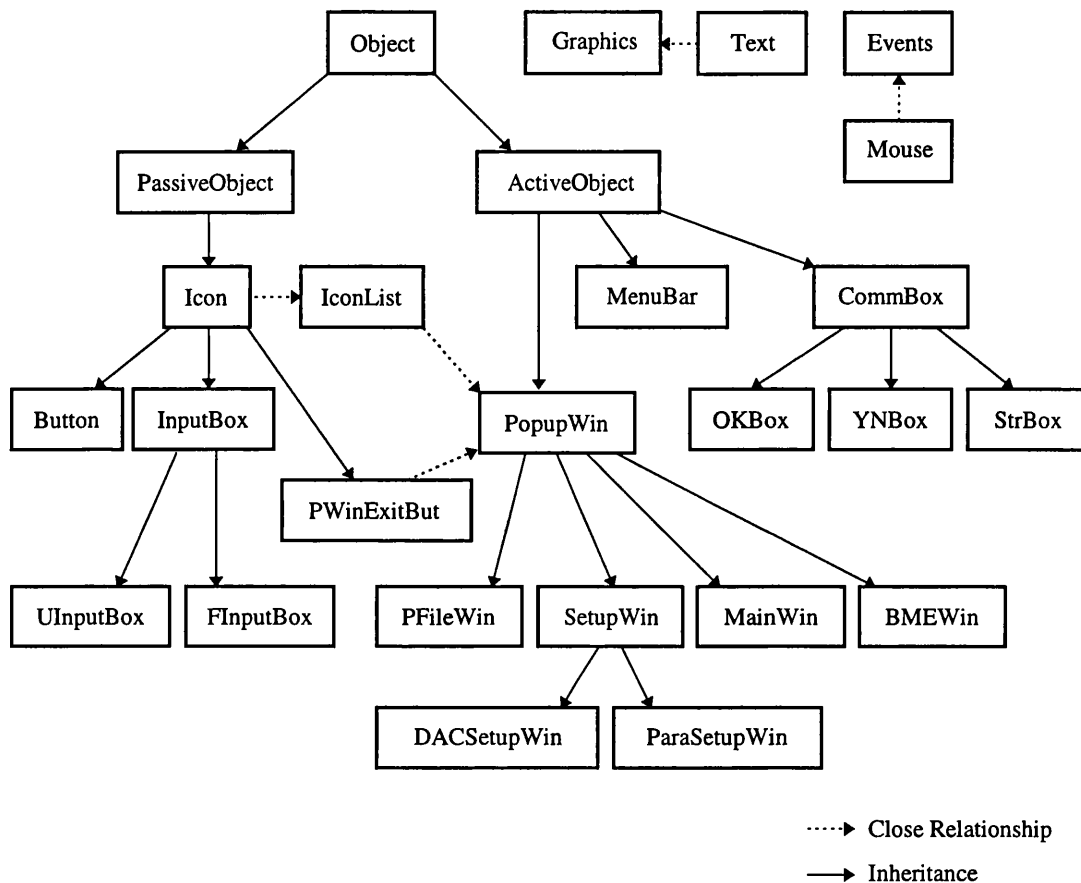


Figure [5.2] Hierarchical tree and relationship of classes for the GUI and control software

To support the graphical activities of other class objects, the Graphics class declares public member functions that access these graphics functions in the library. Its public member functions are invoked through the **extern** variable **GMaster** declared as a pointer to a Graphics object. Therefore, it is needed to create a new Graphics object and assign its address to the **GMaster** variable. The constructor of the Graphics class performs the identification of available system resources such as the main processor, the graphics card, and the amount of video memory. It configures the video card to the

display mode that supports the colour depth of 256 and the resolution of 640×480 pixels. The previous display mode is saved at the construction of a Graphics object and restored at its destruction.

The header file, GRAPHICS.HPP where the Graphics class is declared defines two enumeration data types; BOOL(Boolean) and FLAG. Other data types such as BOX and COLOUR are defined to hold the coordinates of a rectangle on the screen and to hold the information on the display colour, respectively. It also provides a function to handle the dynamic memory allocation error that occurs when the allocation of free memories demanded by the **new** operator fails. The Text class is defined to provide a more convenient way to print texts on the screen in graphical display mode. It has no constructor nor destructor defined, but only three public member functions for three types of text alignment; left, centre, and right. They use one of the member functions of the Graphics class to print texts or strings of characters within the given rectangular boundary that is passed over as an argument. It is also possible to specify the display colours of characters and the background. When the text is too long to fit into the given boundary, no character is printed on the screen to prevent any graphical error.

5.1.3 Classes to Handle User Activities

The class objects in the GUI display the static or dynamic abstraction of the system parameters and accomplish tasks based on the user activities from peripheral devices. The Events class provides public member functions to interpret and handle these user activities from the keyboard and mouse pointing device. It also has member functions to control the mouse. As in the case of the Graphics class, the access to its public member function is made through the **extern** variable, EMaster declared in the EVENTS.HPP header file, which is a pointer to an Events object. The header file also includes the definition of integer values of several useful keyboard inputs such as function keys and arrow keys.

The Events class has an object of the Mouse class defined as a protected data member. The constructor of the Mouse class identifies the available mouse pointing device and defines the shape of its pointer on the screen. Through this data member, an object of the Events class obtains the information on the status of mouse buttons and the

coordinates of the mouse. It also controls the mouse by direct access to the public member functions of the Mouse class. After the identification and interpretation of the user activities, the `ProcessEvents()`¹ member function returns a **struct** data type, `EVE`. It has five integer data members to hold the keyboard input and old and new mouse coordinates. It uses the enumeration data type, `EVEmode` to return the identified user activity.

5.1.4 PassiveObject Class

Being displayed graphically on the screen, many objects in the GUI have common characteristics. For example, they have coordinates and functions to display graphics regardless of their particular functionality. The `Object` class is defined to serve as the base class of these class objects. It provides the most fundamental conceptual foundation to other classes. It has member functions and data members that are most common to its derived classes and serves as the root class or the *abstract superclass* in the inheritance hierarchy. It is a pure abstract base class with graphical member functions declared as pure virtual functions. Two classes, `PassiveObject` and `ActiveObject` are derived directly from the `Object` class to serve as base classes in the inheritance hierarchy. Their objective is to clarify the conceptual definition of many classes derived from the `Object` class and to provide an intermediate stage for the future expansion of the GUI. They are also pure abstract classes because their graphical member functions are inherited from the `Object` class. Therefore, the construction of any object of these three classes is not allowed.

The distinctive difference between two base classes, `PassiveObject` and `ActiveObject` is the existence of member functions, `ProcessEvents(...)` and `Run()`. As its name implies, objects of the classes derived from the `PassiveObject` class are used to represent the static information in caption or the dynamic information based upon the user activities or the result of some tasks. Therefore, the `PassiveObject` class does not provide any member functions to process the user activities or to perform tasks. As can be seen in Figure [5.2], it is the base class of six classes; `Icon`, `Button`, `InputBox`, `UInputBox`, `FInputBox`, and `PWinExitBut`.

¹ Functions without arguments are followed by `()`. Functions with arguments by `(...)`

Presently, the `Icon` class is the only class derived directly from the `PassiveObject` class. It has public member functions to handle graphics and other functions to dynamically update the information displayed on the screen. Because they are not pure virtual functions, the construction of its object is allowed unlike its base class. There are three classes derived from the `Icon` class; `PExitButton`, `Button`, and `InputBox`. A class closely related to the `Icon` class is the `IconList` class. It implements a conventional dynamic linked-list to hold pointers to any `Icon`-derived class objects and provides various member functions to manipulate and access its linked-list. It is defined to assist any `PopupWin`-derived class object in managing its `Icon`-derived class objects. Any `Icon`-derived class object is *included* in a `PopupWin`-derived class object when its pointer is in the linked-list data member, i.e., an object of the `IconList` class. Refer to sub-section 5.1.5 for the description of the `PopupWin` class and its use of the `IconList` class.

The `Button` class is defined to display some static information on the screen or to symbolise various functions of a `PopupWin`-derived class object. Its object can display two strings of characters on the button-like rectangular background. The `InputBox` class is used to reflect the assignment made to a certain variable from keyboard inputs. On the screen, its object consists of two rectangular fields; one on the left for the name of the variable and the other on the right for the value assigned to the variable. Rather than returning the assigned value when it is deactivated, its constructor has the reference argument that is alias to the variable. An `InputBox` class object included in a `PopupWin`-derived class object is activated when its left field is selected and prints the input characters from the keyboard without any discretion. Two more specific classes are derived from the `InputBox` class; `UInputBox` and `FInputBox`. The `UInputBox` class is used to reflect the assignment of an **unsigned int** variable while the `FInputBox` class to reflect that of a **float** variable. In addition to the inherited public member functions, they have functions to convert the verified input characters to integers and floating-point numbers, respectively and vice versa. At the deactivation, if the assigned values are out of the given maximum and minimum, the original values passed over at the activation are restored. Their maximum and minimum are defined when their objects are constructed. The `PWinExitBut` class is defined for the specific purpose to provide the exit button icon to any `PopupWin`-derived class object. It is declared in the protected member area of the `PopupWin` class and therefore is instantiated only by a

PopupWin-derived class object. The constructor of its *parent* object automatically instantiates a PWinExitBut class object and pushes it into the IconList linked-list. One of the constructor arguments of the PWinExitBut class is the pointer to its parent PopupWin-derived class object. When the PWinExitBut class object is activated, Exit(), the member function of its parent object used to terminate its operation, is invoked through this pointer.

5.1.5 ActiveObject Class

The pure abstract base class, ActiveObject derived from the Object class provides public member functions to handle the user activities and to perform various tasks. It has three distinctive member functions; Run(), Exit(), and ProcessEvents(...). Being invoked, the member function, Run() of a ActiveObject-derived class object performs the appropriate tasks in accordance with the user activities identified by the ProcessEvents(...) member function. By executing the function, Exit(), the control of the programme flow is handed back to the object or routines that invoked the Run() function. Three classes are derived from the ActiveObject class; PopupWin, MenuBar, and CommBox.

5.1.5.1 PopupWin Class

Like the Icon class derived from the PassiveObject class, the PopupWin class is the base class for all classes that exhibit the window-like graphics environment. Three public member functions are defined to display and update graphics in the window and to clear its working window area. It has protected data members to define some fundamental properties of the window such as the title and background and text colours. Its object can include any Icon-derived class object. As a data member, it has the pointer to an IconList object, PIconList to hold pointers to Icon-derived class objects in the linked-list. Its constructor defines the fundamental window properties, initialise several parameters, and constructs an IconList object whose pointer is assigned to the PIconList. As described previously, it instantiates a PWinExitBut object and pushes its pointer into the linked-list.

Being derived from the `ActiveObject` class, it executes its tasks through the `Run()` member function. The `Run()` function invokes the `ProcessEvent(...)` function to obtain the identified user activities, according to which it invokes other protected member functions. All of them investigate whether the user activities are made on the subject of any particular `Icon`-derived class object and invoke its corresponding member functions. Each of them has specific routines to perform the assigned tasks. At the completion of its tasks, the `Run()` function resigns the control of the programme flow and the management of the user activities by invoking the `Exit()` function. Pressing the right mouse button automatically invokes the `Exit()` function. Three application-dependent classes are derived from the `PopupWin` class; `BMEWin`, `MainWin`, and `SetupWin`. The `PFileWin` class is also derived to assist the file operation. The description of their functions are presented in the following sections.

5.1.5.2 MenuBar Class

An object of the `MenuBar` class is used to represent and control a menu bar that can be included in a `PopupWin`-derived class object. In addition to the `MenuBar` class, the `MenuFrame` class is declared in the `MENU.HPP` header file. Its object is used to define and display each sub-frame of the menu bar. An object of the `MenuBar` class has the pointer data member to the `MenuFrame` class object. Its constructor requires the number of sub-frames and strings of characters that represent the top level of the menu bar as its arguments. It allocates free memories to hold the array of the given number of sub-frames and assigns the address to the sub-frame pointer. Each sub-frame must be constructed and added to the `MenuBar` class object with the `MakeMenu(...)` member function. It has arguments for the number of items in the sub-frame, strings of characters for the items, and optional strings of characters for the short keys.

Because the `Run()` member function inherited from the `PopupWin` class is declared to have no arguments and to return a `BOOL` data type, The `MenuBar` class provides the overloaded member function `Run(...)`. It has two reference arguments of `int` data type to return the identities of the sub-frame and its item selected by the user. It is the responsibility of the object or routines that invoke this function to interpret these two return values and to perform the appropriate tasks. Unlike any `PopupWin`-derived class

object, an object of the MenuBar class stores the background image when a sub-frame is opened and restores it when the selection is made.

5.1.5.3 CommBox Class

To communicate with the user during the execution of the GUI, four classes are defined; CommBox, OKBox, YNBox, and StrBox. The CommBox class is the base class of others and provides common members including the pointer to an IconList class object. Its object is used to prompt a message for a short duration of time. The OKBox class is used to notify the user the occurrence of any event that needs the user's attention and acknowledgement. The YNBox class is used to prompt any message and to make a suggestion that the user either accept or discard using two Button class objects. When the GUI needs a string of characters from the user, an object of the StrBox class can be used to retrieve it from the keyboard. It overloads the inherited Run() member function to return the input string of characters. As for the MenuBar class, objects of these four classes store the background image to display their windows and restores it when the communication is completed.

5.2 Bitmap Editor

5.2.1 Overview

As explained earlier, the need for a graphical Bitmap Editor arose from the author's past experience. The Bitmap Editor is an object of the BMEWin class that is derived from the PopupWin class. It represents the Bitmaps graphically with streams of pulses on the screen and thus provides the solid visualisation of Bitmaps for the user. The user can easily define, examine, and modify the Bitmaps with this invaluable instrument. The Bitmap Editor is integrated in the GUI through the MainWin class object as described in section 5.3.

The main functions of the Bitmap Editor are to manage Bitmap files and to manipulate the Bitmaps. The BMEWin class has member functions to display the Bitmaps and the names assigned to Bitmap channels (node names). It has protected data members to define several parameters such as the maximum number of Bitmap

resolution units, the maximum number of Bitmap nodes or channels, and the maximum and minimum factors of display magnification. They are initialised by its constructor. As described in Chapter 4, the system hardware limits the number of Bitmap resolution units to 1024 and the number of Bitmap channels to 24. Two protected data members are used to store the Bitmaps and node names in one-dimensional arrays. The BMEWin class also has the pointer to a MenuBar class object as its data member. Its constructor creates a MenuBar class object and defines three sub-frames. One member function is assigned to each item in each sub-frame to perform the appropriate tasks in accordance with the user selection from the menu bar.

5.2.2 Management of Bitmap Files

5.2.2.1 PFileWin Class

The PFileWin class derived from the PopupWin class is defined to provide a simple way to perform operations concerned with files. When activated by invoking the Run(...) member function, its object presents a dialogue box with which the user can select the desired file in a sub-directory. It displays all file names that have the given extension within the sub-directory sorted in alphabetic order. Its overloaded member function, Run(...) has two arguments to receive the name of the sub-directory to be searched and the extension of the file name. It returns the full path of the selected file as a string of characters. The PFileWin class also supports the search for files within different sub-directories and accepts the file name typed in by the user.

5.2.2.2 File Functions

The Bitmap Editor stores all Bitmaps as streams of text characters in files that have the extension, BIT. Therefore, the user can edit or examine any Bitmap file with the standard text editor if desired. Each Bitmap file contains the number of nodes, the number of Bitmap resolution units, and the stream of Bitmaps written line by line in hexadecimal format and followed by numerical indices. The numbers of Bitmap nodes and resolution units are used by the Bitmap Editor to dynamically allocate free memories to the arrays that store the Bitmaps and node names. A portion of the horizontal transfer Bitmap file used in the testing of the control system is shown in Figure [5.3].

The Bitmap Editor begins its operation by reading its initialisation file, BME.INI. In case of its very first operation or the failure of the access to BME.INI, a new initialisation file is automatically generated in the current working sub-directory. The initialisation file contains the name of the sub-directory that Bitmap files reside in and the name of the Bitmap file accessed most recently. The Bitmap file operation is carried out through the menu bar or the short keys. The File menu has the sub-frame items such as New, Open, Save, and save As, each of which has its own short key. The New item creates a new Bitmap file named UNTITLED.BIT. Its numbers of Bitmap nodes and resolution units are given by the user and all Bitmaps are initialised LOW. The functions of other items are obvious in their names. Open and save As items get the name of a Bitmap file from the user by constructing a PFileWin class object and invoking its Run(...) member function. The Bitmap Editor is shown in Figure [5.4] when the Open item is selected.

24	16			
NA		NA	PC3-PD3	PC1-PD1
PC2-PD2		PA1-PB1	PA3-PB3	PA2-PB2
RGTR-RGBL		S2T	S1RT-S1RB	S3RT-S3RB
S2B		S3LT-S3LB	S1LT-S1LB	RGTL-RGBR
M_CLK1		S/H1A	S/H1B	ADC1
M_CLK2		S/H2A	S/H2B	ADC2
0	c0dd00			
1	c0dd00			
2	c05c00			
3	c05c88			
4	c05c88			
5	c05c88			
6	c01488			
7	c01488			
8	c036aa			
9	c036aa			
10	c02288			

Figure [5.3] A portion of the horizontal transfer Bitmap file
The clock timing pulses for this portion are shown in Figure [6.8.b].

The current version of the Bitmap Editor has no function to examine whether the contents of a Bitmap file have been modified since it was opened. Therefore, the attention of the user is necessary to ensure the Bitmap file is saved before a new

Bitmap file is created or a different file is opened. When a Bitmap file is modified by other text editors, the user must make it certain that the numbers of Bitmap nodes and resolution units defined at its head agree with those of actual nodes and Bitmaps. The disagreement between them will cause the Bitmap Editor to hang up because of the improper management of free memories.

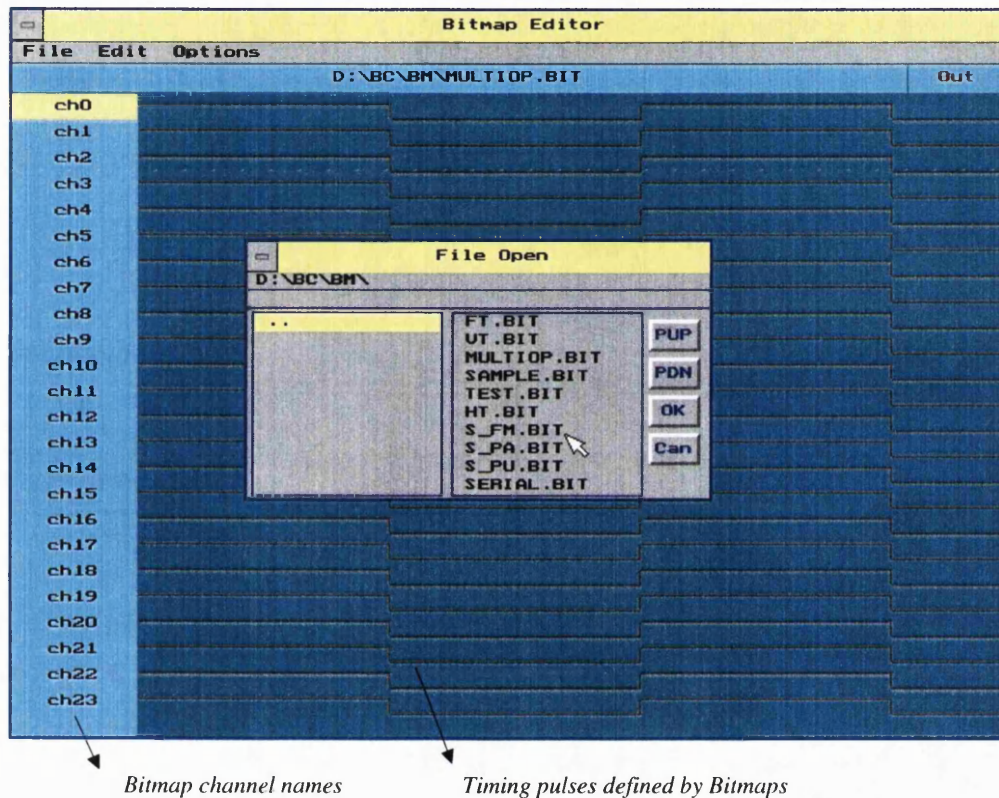


Figure [5.4] Bitmap Editor when the Open item in the File menu is selected

5.2.3 Manipulation of Bitmaps

To provide a flexible way to define and modify the Bitmaps, the Edit menu of the Bitmap Editor has five elementary items; HighBM, LowBM, Copy, Paste, and Node. HighBM and LowBM items are used to assign HIGH and LOW states to the pulse streams of a Bitmap channel, respectively. With Copy and Paste items, a portion of the pulse streams of any Bitmap channel can be copied into the buffer and pasted to other pulse streams. All these Edit items require that a particular portion of the pulse streams of

any Bitmap channel is selected with the mouse. For the Paste item, only the beginning of the selected portion is effective regardless of its length because the length is given by the portion to be pasted. With the Node item, the node name assigned to each Bitmap channel can be modified. Not being used by the control software, it assists the user in understanding the physical meaning associated with each Bitmap channel.

5.2.4 Miscellaneous Functions

The Options menu provides several other items. The Home item displays the Bitmaps from the beginning whereas the Goto item does from the given index of Bitmaps. With the Size item, the user can modify the length of Bitmaps. If the new length is shorter than the current one, the Bitmaps are truncated at the end. In the other case, more Bitmaps initialised LOW are added at the end of the original Bitmaps. There are several other miscellaneous functions provided by the Bitmap Editor. As well as by clicking the mouse button on node names, the user can change the selected node by using the up and down arrow keys. The right and left arrow keys move the Bitmaps displayed on the screen by one unit of the magnification factor to right and left of the window, respectively. The page up and down keys are used to change the factor of magnification from 1 to 24. The name of the current Bitmap file is displayed below the menu bar and the index of Bitmaps at the current position of the mouse is displayed next to the Bitmap file name. This information can be used to select the desired portion of Bitmaps accurately. At the occurrence of any event that requires the user's attention, the appropriate message is prompted at the bottom of the window.

5.3 Main Control Routine

5.3.1 Overview

All class objects and routines for the GUI and control software are integrated in a MainWin class object. The main() function of the control software constructs Graphics and Events class objects and assigns their addresses to GMaster and EMaster **extern** pointer variables. Subsequently, it constructs a MainWin class object and begins the GUI by invoking its Run() member function. The MainWin class is another application-

specific class derived from the PopupWin class. It has three pointer data members to PopupWin-derived class objects. One of them is used for the Bitmap Editor while others for the set-up of DACs and other parameters of the control system. Its constructor instantiates a Button class object that corresponds to each pointer. In addition, it provides public member functions to perform control and auxiliary tasks such as error checking, writing the system parameters and Bitmaps, readout of the CCD, and display of image frames. Button class objects are also created for these member functions.

5.3.2 Set-up of DACs and Other System Parameters

As described earlier, the system hardware provides numerous system parameters to be controlled. To reflect the static DAC control method and the sequencer's ability to drive CCDs of different physical formats, two application-specific classes, DACSetupWin and ParaSetupWin are defined. They are derived from the SetupWin class, which is again derived from the PopupWin class. An object of any class derived from the SetupWin class is expected to accommodate a number of InputBox or InputBox-derived class objects in its IconList object. Each of them is used to set up a particular parameter. To manage a large number of InputBox or InputBox-derived class objects effectively, the SetupWin class has the pointer data member, OldIcon to the Icon class object. It is used to hold the address of the Icon class object accessed most recently by the user.

The DACSetupWin class object constructs FInputBox class objects to specify the output voltage levels of all octal DACs on Bias and two Clock Boards. Eight FInputBox class objects are created for the octal DAC on the Bias Board and 32 are created for four octal DACs on two Clock Boards. Several **extern** variables are declared in the header file, SETUPWIN.HPP for the name of each DAC, its output voltage level, and the allowed maximum and minimum voltage levels. Not being protected data members, any other routines or class objects declared and defined out of the scope of the DACSetupWin class can access them. All of them are defined and initialised in the source file except for those for the output voltage levels. They are initialised by the constructor of the DACSetupWin class with default values defined in two text files,

BIAS.DFT and CLOCK.DFT. The BIAS.DFT file defines the default values of eight bias DACs whereas the CLOCK.DFT file defines those of all clock DACs.

These **extern** variables are passed to the constructors of FInputBox class objects as arguments. The modification of the output voltage level of a DAC is made by activating the FInputBox class object assigned to it and typing the new floating-point value in. Any value that falls out of allowed maximum and minimum is discarded by the FInputBox class object. Figure [5.5] shows the DACSetupWin object displayed on the screen when the FInputBox class object assigned to one of LOW clock DACs is activated.

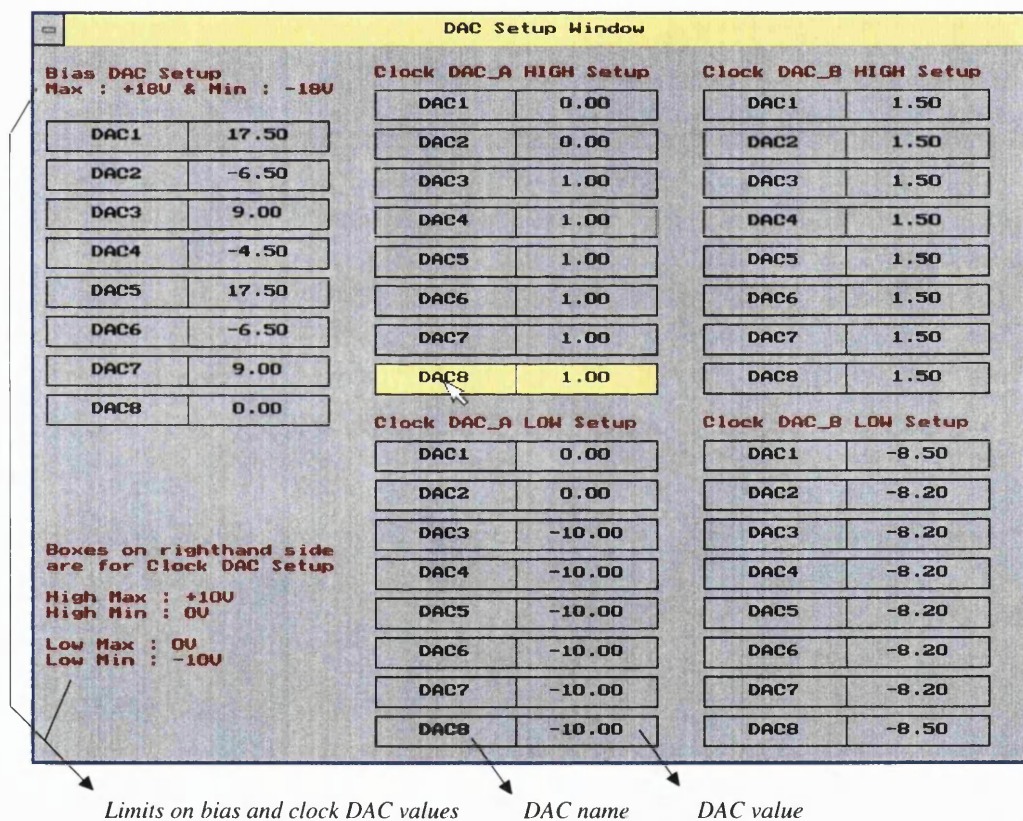


Figure [5.5] Modification of the output voltage level of a LOW clock DAC

The set-up of other system parameters are carried out in the similar manner to that of the output voltage levels of octal DACs. The ParaSetupWin class object constructs five UInputBox class objects to specify the number of repetitions of frame-transfer Bitmaps,

the numbers of horizontal and vertical transfers, the number of image frames to be acquired, and the integration time. It also creates four `InputBox` class objects for four `Bitmap` file names used to read out the CCD; frame-transfer, vertical transfer, horizontal transfer, and purging `Bitmap` files. It uses several **extern** variables and the default file, `PARA.DFT` in the same manner as the `DACSetupWin` class object.

5.3.3 Other Member Functions

The `MainWin` class has four public member functions to perform the control of the system hardware and other supplementary tasks; `Error()`, `Write()`, `Read()`, and `Image()`. They are invoked by activating four `Button` class objects created by its constructor. The `Write()` member function initialises and configures the interface card on the host PC and transmits the various system parameters to the system hardware. Normally, the transmission is carried out in the order of DNs for video offset DACs, DNs for the octal DAC on the Bias Board, DNs for octal DACs on two Clock Boards, the number of image frames to be acquired, the number of repetitions of frame-transfer Bitmaps, the numbers of vertical and horizontal transfers, the numbers of resolution units of four streams of Bitmaps, four streams of Bitmaps to each `Bitmap RAM`, and finally the top address for the `Image RAMs`. The `Read()` function initiates the CCD readout procedure and the `Image()` function retrieves the image frames stored in the `Image RAMs` and displays them on the screen.

The `Error()` function verifies the validity of some system parameters and the Bitmaps given by the user. Whenever an access is made to `DACSetupWin` and `ParaSetupWin` class objects or the `Bitmap Editor`, the commencement of the CCD readout procedure is prohibited unless the `Error()` function is executed. The elements checked are the voltage swings of CCD clock pulses, the existence of the given `Bitmap` files, the lengths of Bitmaps, and the existence of alternating bits in the Bitmaps. The voltage swings of CCD clock pulses given by the difference of the output voltage levels from LOW and HIGH DACs are not allowed to exceed 15 V. This limitation is imposed by the recommended operating conditions of MOSFET drivers on two Clock Boards. In case any of them exceeds 15 V, an error message is prompted to the user for acknowledgement. The `Error()` function also reads all streams of Bitmaps from the

Bitmap files specified by the user, checks their lengths and the existence of any alternating bits in them, and prompts warning messages if necessary. The switching frequency of 10 MHz of the MOSFET driver limits the minimum length of Bitmaps to 8. Its incapability to support switching input pulses with low duty cycles makes it necessary to check any alternating bits in the Bitmaps for the reliable operation of the control system. Refer to section 6.2 for the explanation on this necessity.

System Development, Integration, and Testing

The high-speed CCD control system incorporates the Bitmap and static DAC control methods and provides various system parameters to be controlled by the user. It is also designed to have the capability to drive high-speed CCDs of different formats. The system hardware consists of nine modules and the host PC and each module board is defined according to its distinctive functionality. To assist the user in controlling many system parameters, the GUI that includes the control software is designed based on many class objects. To verify the functionality of the control system and to evaluate its characteristics, it is essential to perform various tests with the complete system. The testing of the system was performed in ‘down-top’ and modular fashion. For the system hardware, following the independent functional verification of prototype module boards, they were integrated and tested one by one. Similarly, for the GUI and control software, the functionality of each class object was verified and each module of routines was tested. Finally, the complete control system was integrated and tested.

In section 6.1, the application-dependent CCD Board is described. In section 6.2, the modification of the system hardware and the limitation imposed by the system hardware are discussed. The integration of the complete control system is briefly introduced in section 6.3. The tests performed to evaluate the control system are explained and their results are presented in section 6.4. Finally, in section 6.5, some possible improvement of the control system is discussed.

6.1 CCD Board

Initially, it was considered to put the CCD and accompanying circuits on the same hardware module with clock drivers to reduce the high-frequency switching noise generated over the transmission of clock pulses. Not only the physical space required to accommodate clock drivers made this approach unfeasible, but it also violated the design philosophy to build a versatile control system. Because the system hardware is designed to have a dedicated CCD board, another CCD board only needs to be manufactured for a different CCD. Instead, to minimise the high-frequency switching noise, miniature coaxial cables are used to transmit clock pulses from two Clock Boards.

Unlike other system hardware modules whose components are mostly wire-wrapped, the CCD Board is a PCB(Printed Circuit Board) that houses the Loral 64×64 frame-transfer CCD and four pre-amplifiers. Four output video signals from the CCD are amplified independently on the CCD Board and transmitted to two Signal Processing Boards via coaxial cables. It has a coplanar ground plane, 14 coaxial cable connectors for clock pulses, two pin headers for biases, and 4 coaxial cable connectors for the amplified video signals.

The Loral CCD has 12 clock gates to transfer the signal charges in image and storage sections and 14 gates in horizontal registers. Each Clock Board can provide up to eight streams of clock pulses, which correspond to byte-wide streams of Bitmaps. To drive 26 clock gates with 16 available clock drivers, therefore, most gate pins of the Loral CCD are tied up in pairs on the CCD Board as shown in Table [6.1]. As the result, six streams of clock pulses are needed to drive vertical clock gates and eight to drive horizontal ones. One of two Clock Boards is assigned to vertical clock gates and the other to horizontal ones as described in Chapter 4. It leaves two Bitmap channels unused on the Clock Board assigned to vertical clock gates. One of them was used to control the integration time in the mean-variance test as explained in section 6.4. The bias pins located at the right and the left of the CCD are also tied up except for the substrate pin. It gives 7 biases altogether, which means one of eight biases provided by the Bias Board is not used.

Biases	Clock gates for image and storage sections	Clock gates for horizontal registers
RDTL - RDBL	PA1 - PB1	RGTR - RGBL
ODTL - ODBL	PA2 - PB2	S2T
OGTL - OGBL	PA3 - PB3	S1RT - S1RB
RDTR - RDBR	PC1 - PD1	S3RT - S3RB
ODTR - ODBR	PC2 - PD2	S2B
OGTR - OGBR	PC3 - PD3	S3LT - S3LB
SUB		S1LT - S1LB
		RGTL - RGBR

Table [6.1] Pin configuration of the Loral CCD on the CCD Board

Initially, to ensure the charge transfer in the correct direction for the Loral frame-transfer CCD, clock gates of the upper image section were tied to those of the lower image section and clock gates of the upper storage section with those of the lower storage section. However, after its delivery, it was revealed that the image section of the Loral CCD is not masked. It means that the CCD is eventually a full frame CCD that has six independent vertical clock gates for each half. Consequently, the pin configuration had to be modified to reflect this unusual characteristic and to drive it as a full frame CCD.

The video signal from each CCD output node is coupled by a single-stage emitter follower and a simple RC filter. The emitter follower provides a high input resistance to the CCD output node and has a wide bandwidth. The RC filter eliminates the high DC pedestal from the video signal. The AC-coupled video signal is amplified by the following non-inverting amplifier and transmitted to one of two Signal Processing Boards via a coaxial cable. The feedback resistor of the op-amp is selected to give the bandwidth of approximately 30 MHz, which is three times the maximum pixel sampling frequency, 10 MHz. The circuit diagram of the pre-amplifier for each output video signal is shown in Figure [6.1]. In this diagram, R1 is the current load to the CCD output transistor.

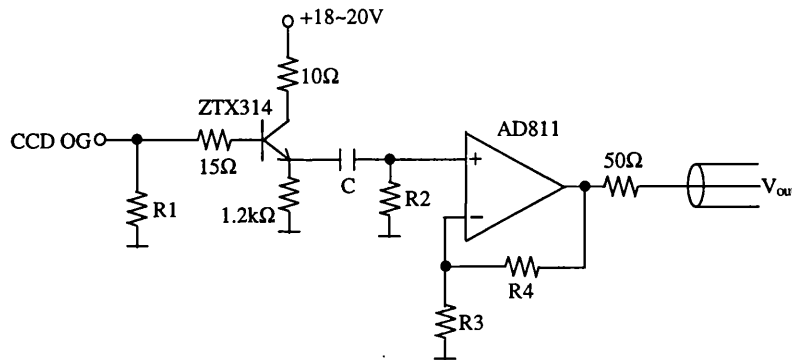


Figure [6.1] Circuit diagram of the AC-coupled pre-amplifier

6.2 Modification of the System Hardware

6.2.1 Reduction of the Switching Noise

In the beginning, streams of Bitmaps generated by the Bitmap Generation Board were transmitted to Clock Boards and Signal Processing Boards via single wires running over the back of the system sub-rack. It was pointed out that this arrangement caused the coupling of the high-frequency switching noise to other module boards as well as being susceptible to the external noise. After transmission, high ringing and over- and under shoots were also observed on pulse streams, which is thought to have been caused by the impedance mismatch between the two transmission nodes. As described in Chapter 4, to reduce the coupling of the switching noise, high-speed CMOS differential line drivers and receivers are employed for the transmission of streams of Bitmaps. The differential streams of Bitmaps are transmitted over 100 Ω twisted pairs terminated at the receiving node to eliminate the impedance mismatch between the transmission nodes.

Even after the implementation of differential line drivers and receivers, high over- and under-shoots persisted over TTL pulse streams of Bitmaps restored on two Clock Boards. They were observed to originate from the Bitmap Generation Board prior to the transmission. Some modification was made on the Bitmap Generation Board to reduce the generation of the high-frequency switching noise. Serial resistors were added between the sequencer and Bitmap RAMs as well as between Bitmap RAMs

and differential line drivers. This method of impedance matching with serial resistors was found effective and required the least physical space and amount of modification on the Bitmap Generation Board. The resistances were chosen experimentally. Wherever it was practical, this method was employed between devices that operate in high speed over the entire system hardware. In addition, a simple Schottky diode circuit was added to the output node of the master clock on the Bitmap Generation Board to limit its output voltage swing and to produce a smoother transition. It was observed to contribute to the reduction of over- and under-shoots on the output signals from the sequencer.

To realize streams of clean clock pulses on Clock Boards, combinations of simple RC filters were tried between MOSFET drivers and CCD clock gates. Especially, a ‘snubber’ filter and a serial resistor with a small parallel capacitor over the transmission line were found effective¹. On the other hand, however, they caused an increase of rise and fall times of CCD clock pulses. The simplest but effective solution was to put a small serial resistor of about 10 Ω at the positive power supply of the MOSFET driver as shown in Figure [4.11]. Unlike other circuits, it did not cause the considerable increase of rise and fall times of CCD clock pulses.

6.2.2 CCD Clock Drivers

Being inherently capacitive, CCD clock gates require clock drivers that have the capability to supply high currents in high-speed operation. The switching frequency of clock drivers also dictates the horizontal transfer time that is dominant in the total readout time. Therefore, along with the sequencer, they have a huge influence on the performance of the control system in terms of the readout speed. Initially, various simple circuits made of discrete devices were tried as clock drivers. However, the relatively long turn-off time of discrete devices and the stray capacitance associated with the prototype boards made them inapplicable to this work.

Instead, the MOSFET driver, EL7212 was tried as the clock driver. It can supply a peak current up to ± 2 A and shows typical output rise and fall times less than 10 ns

¹ The ‘snubber’ filtering is a parallel termination with a serial combination of resistor and capacitor.

with a 500 pF load at the voltage swing of 15 V. Its rise and fall times were ideal for this work taking the Bitmap resolution unit of 12.5 ns into account. However, its maximum switching frequency is limited to 10 MHz when driven continuously. This characteristic was not considered to limit the performance of the control system in terms of the readout speed because 10 MHz satisfies the required frame rate. Figure [6.2] shows the circuit diagram of the clock driver built with EL7212. In this diagram, the high-speed op-amp, AD811, drives the MOSFET driver using the TTL output from the differential line receiver. It translates the TTL output level by the amount of the negative power supply to EL7212, which is defined by one of LOW DACs. In other words, it gives output pulses with the approximate voltage swing of -LOW DAC O/P to -LOW DAC O/P + 4 V. The EL7212 generates CCD clock pulses switching between its power supplies defined by one of HIGH DACs and one of LOW DACs.

During the testing of Clock Boards, however, the EL7212 exhibited high heat build-up and was often found to be damaged. When damaged, it drew a substantial amount of current from its power supplies and subsequently damaged the neighbouring devices. It is believed either that the EL7212 cannot operate with a negative power supply or that its input pulse was not properly referenced to the negative power supply. The EL7212 is described in the data book to be pin-to-pin compatible with the more widely used MOSFET driver, DS0026 that can operate with a negative power supply. It is believed that it cannot operate with a negative power supply, which can be justified by the fact that the EL7212 driving the reset gate of the CCD output transistor never failed. The nominal voltage level of the reset clock is 0 to +8 V. It was also found during the testing that the EL7212 is unable to support input pulses with low duty cycles. For example, when a channel of Bitmaps has the alternating format of either 10111111... or 01000000..., it failed to generate the desired output pulse.

All EL7212's were replaced with DS0026's and Clock Boards were subsequently modified as shown in Figure [4.11]. The DS0026 has specifications slightly inferior to EL7212. It is described to give rise and fall times of 20 ns with a 1000 pF capacitive load and supply a peak current of ± 1.5 A. Its maximum switching frequency lies between 5 and 10 MHz depending on power dissipation. Because the capacitance of horizontal clock gates that are switched at the highest speed is relatively small, it was believed that the DS0026 would not give rise to degradation of the system

performance. For instance, the horizontal clock gate of the 512×512 full frame CCD from Loral Fairchild has a capacitance of 150 pF[1]. The horizontal clock gate of the Loral CCD used in this work is expected to have smaller capacitance because it is a function of the physical size of a CCD. The adequacy of DS0026 as the clock driver was verified by driving it continuously at 10 MHz with a 150 pF load and the nominal horizontal clock voltage swing of 10 V. It exhibited rise and fall times less than 12.5 ns. However, it also failed to generate the desired output pulse when the input pulse has a low duty cycle.

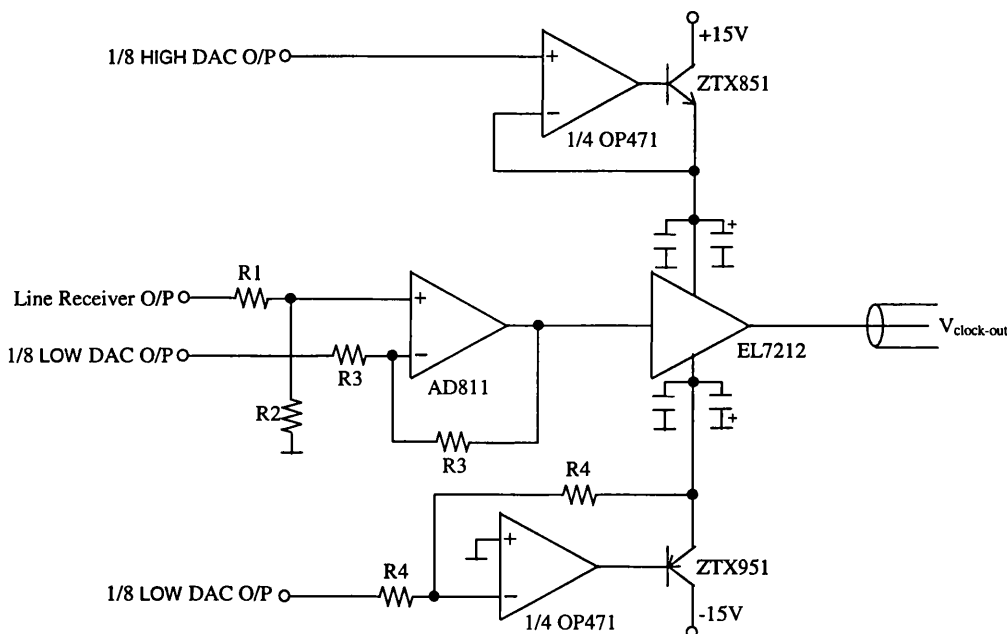


Figure [6.2] Circuit diagram of the former clock driver built with the high-speed MOSFET driver EL7212

6.3 Integration of System Modules

All prototype module boards of the system hardware except for the CCD Board were constructed with standard Eurocard boards, each of which has a coplanar ground plane. All connections between components on each module board were wire-wrapped whereas the wires for power supplies were soldered. Each module board was assembled on a universal sub-rack. Two separate bus bars along the back of the sub-

rack are used to distribute power supplies provided by the main power supply unit and the Voltage Regulation Board. The top bus bar distributes all analog power supplies of $\pm 18 \sim \pm 20$, ± 15 V, and ± 5 V together with the analog ground return and the bottom one distributes the digital power supply of $+5$ V and ground return. The obvious reason to employ separate digital and analog ground returns is to minimise the coupling of the high-frequency switching noise.

To save the physical space, miniature coaxial cables are used to transmit streams of clock pulses from two Clock Boards to the CCD Board and video signals amplified on the CCD Board. Twisted pairs are used for the 16-bit wide data bus from the interface card on the host PC to the Bitmap Generation Board and for the byte wide bus from the Signal Processing Board #2 to the interface card. Figure [6.3] is the photograph of the complete control system with the main power supply unit and the host PC. The GUI runs on the host PC and several image frames acquired from the Loral CCD are displayed on the host PC monitor in this photograph.

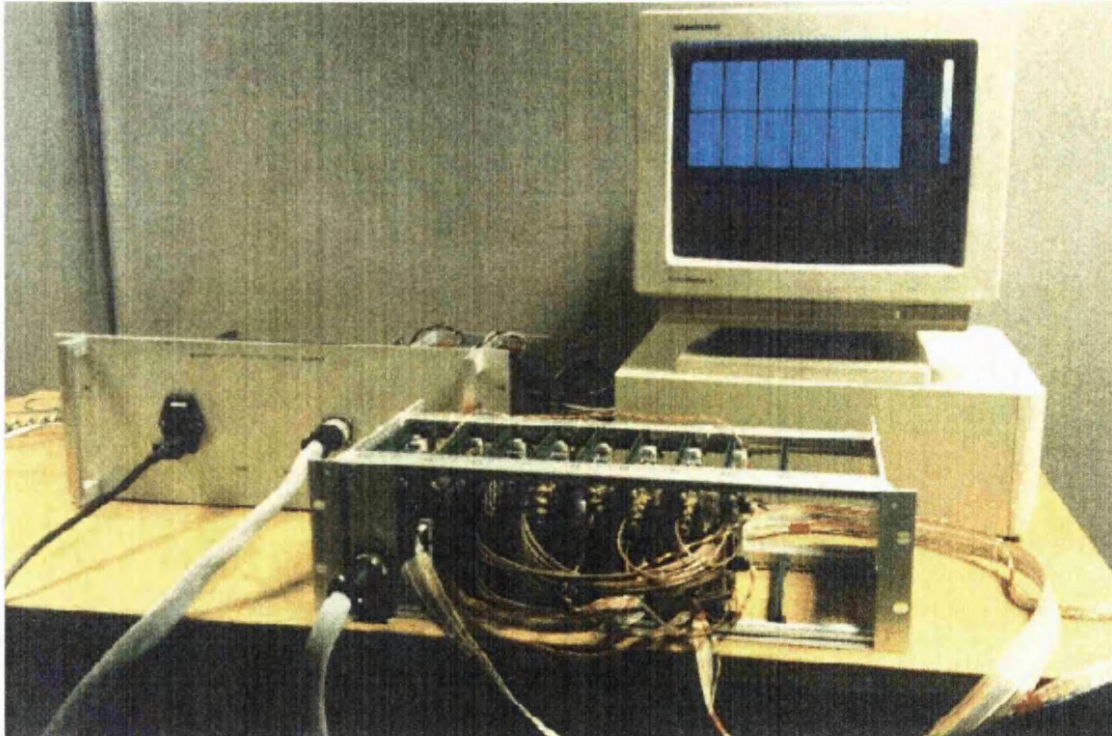


Figure [6.3] Photograph of the complete control system with the main power supply unit and the host PC. Three consecutive image frames acquired from the Loral CCD are displayed on the screen.

6.4 Performance of the Control System

6.4.1 Functional Verification

As described earlier, the verification of the functionality of the control system was performed in 'down-top' and modular fashion. Both in the course of and after the integration of system modules, each module was tested to verify its functionality. For example, the system hardware was tested and integrated in the order of Voltage Regulation Board, Bitmap Generation Board, Bias Board, Clock Boards, Signal Processing Boards, and CCD Board. For the GUI and control software, all classes were tested with their objects in the order of Graphics, Events, Icon-derived classes, PopupWin-derived classes, MenuBar, and CommBox-derived classes. Other components were tested without any particular order. Some verification made in the functional testing procedure is itemised below.

- Operation of the Bitmap Editor
- Operations of DAC set-up routines in the GUI
- Operation of other control software routines
- Transmission of DNs for the physical format of the CCD, four sets of streams of Bitmaps, and DNs for DACs from the host PC via the interface card
- Generation of streams of Bitmaps by the sequencer in correct order for the given number of times
- Output voltage levels of DACs assigned to CCD biases in agreement with their DNs
- Output voltage levels of octal DACs assigned to CCD clock gates in agreement with their DNs
- Generation of CCD clock pulses as defined by streams of Bitmaps in correct order with the specified voltage swings
- Output voltage levels of DACs assigned to the DC video offsets in agreement with their DNs
- Generation of sampling signals for S/H amplifiers, conversion signals for ADCs, and write ENABLEs for Image RAMs as defined by streams of Bitmaps and delay lines

- Generation of streams of Bitmaps in agreement with the given number of image frames to be acquired and the readout mode
- Control of the address bus to the Image RAMs both in the CCD readout procedure and in the retrieval process of the pixel values
- Transmission of multiplexed pixel values stored in the Image RAMs to the host PC
- Reception of the transmitted image frames and their display on the screen

6.4.2 Image Acquisition Test

When the functional verification of the control system was completed, the image acquisition from the Loral CCD was performed. Because no recommended operating conditions were available from the manufacturer, two sets of operating conditions were referred to in order to define bias and clock voltage levels of the CCD. The first one is that of commercial full frame CCDs manufactured by the Loral Fairchild[20], which is given in Table [6.2]. Most of them have similar recommended operating conditions regardless of their physical size or architecture. The second one is the test conditions used by the Institute of Astronomy in the University of Cambridge[73] to drive the same CCD used in this work, which is shown in Table [6.3].

The test conditions in Table [6.3] were used for the image acquisition except for the voltage swing of the vertical clock gates. The voltage swing of -10.0 to +1.0 V instead of -12.5 to +1.0 V was used because the Clock Board is unable to supply LOW clock voltages lower than -10.0 V. The CCD was covered with a opaque mask that has a narrow vertical slit along the direction of the vertical charge transfer. The streams of clock pulses similar to those shown in Figure [6.8] were used to drive the CCD. They are MPP clock pulses and the signal charges is integrated in the implant region under the phase 3 of the vertical clock gates. The pixel sampling frequency is 5 MHz, i.e., the length of the horizontal transfer Bitmaps is 16. The vertical transfer time is 1.2 μ s, i.e., the length of the vertical transfer Bitmaps is 96. As pointed out in Chapter 4, the horizontal clock gates were switched over the vertical transfer period to prevent the external capacitors at the input nodes of MOSFET drivers from discharging. The simultaneous sampling of the video signal by two S/H amplifiers during the frame

transfer eliminates the gradual decrease of their output voltage levels, which is caused by the discharge of their internal sampling capacitors. An LED(light-emitting diode) was used as the light source to integrate photons during the frame transfer.

Figure [6.7] shows two sequences of image frames acquired from different combinations of CCD output nodes in the burst readout mode. Figure [6.7.a] shows four consecutive image frames acquired when the signal charges were sampled at the bottom two output nodes. They show the image of the vertical slit in light blue and the overscan region in light purple. The display palette was adjusted to enhance the appearance of the image frames. Two other output nodes were also sampled and the image frames from them have the similar colour level to the overscan region. Figure [6.7.b] shows eight image frames acquired through all four output nodes. The CCD was also driven both at the pixel sampling frequency of 10 MHz and with the MPP operating conditions given in Table [6.2]. The control system gave the image frames similar to those in Figure [6.7] in both cases.

Bias voltage levels (V)		Voltage levels for clock gates (V)		
V_{DD}	20.0		MPP operation	non-MPP
V_{RD}	13.0	Vertical Clock	-5.0 ~ +5.0	0.0 ~ +16.0
V_{OG}	1.0	Horizontal Clock	-8.0 ~ +2.0	0.0 ~ +14.0
V_{SS}	0.0	Reset Pulse	0.0 ~ +8.0	0.0 ~ +12.0

Table [6.2] Recommended operating condition of most commercial full frame CCDs from Loral Fairchild

Bias voltage levels (V)		Voltage levels for clock gates (V)	
V_{DD}	17.5	Vertical Clock	-12.5 ~ +1.0
V_{RD}	9.0	Horizontal Clock	-8.5 ~ +1.5
V_{OG}	-6.5	Reset Pulse	-8.2 ~ +1.5
V_{SS}	-4.5		

Table [6.3] Test operating condition of the Loral CCD from the Institute of Astronomy in the University of Cambridge

6.4.3 Linearity and Mean-Variance Tests

The response of the Loral CCD and the control system to the linear variation of the incident light intensity was examined with an LED light source. The light intensity was varied by controlling the amount of the current flow through the LED with a variable serial resistor. The CCD was constantly exposed during the entire readout period and the integration of the incident photons was made over the frame-transfer period. It was assumed that the contribution of photons integrated over the vertical and horizontal transfer period to the output video signal is insignificant because of the relatively short readout time. The horizontal clock gates were driven at 5 MHz (16 Bitmap resolution units) and the vertical ones at 833.3 kHz (96 Bitmap resolution units). For each discrete variation of the light intensity, several image frames were acquired through the bottom left output node. The pixel values over a sub-area in one of them were averaged to obtain their mean value. Figure [6.4] shows the mean pixel values with open circles in ADU(ADC unit) against the amount of the current flow through the LED. The solid line shown in this figure was acquired by least-square fitting the measured data points.

To prevent the CCD from being exposed over the vertical and horizontal transfer period, in another experiment, one of two Bitmap channels unused on the Clock Board assigned to the vertical clock gates was used to define the on/off pulse to the LED light source. The length of the frame-transfer Bitmaps was fixed and their number of repetitions was varied by the given unit to control the exposure time. In this case, the exposure time was given by the number of repetitions of frame-transfer Bitmaps times their length. Figure [6.8] shows three streams of clock pulses used in this experiment; frame, vertical, and horizontal transfers. It can be seen in Figure [6.8.a] that the top Bitmap channel used to define the on/off pulse to the LED is HIGH over the entire frame-transfer period.

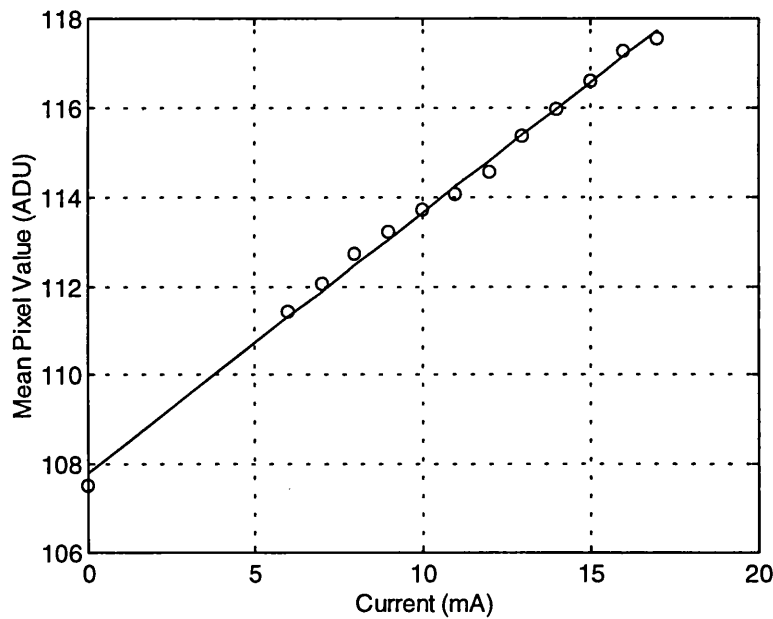


Figure [6.4] Response of the Loral CCD and the control system to different exposure levels. The open circles are the measured data points and the solid line is the least-square fitted line of them.

At each exposure time, three consecutive image frames were acquired to quantitatively analyse the characteristics of the control system, such as the gain and the overall readout noise. The alternative and simple mean-variance test method described in Appendix C was used for the analysis. Figure [6.5] shows the mean pixel values (S_m) with open circles in ADU against the frame-transfer time (exposure time) over small sub-areas from one of the image frame sequences. The length of the frame-transfer Bitmaps was 256 and their number of repetitions was increased from 128 to 2048 by 128 (from 0.41 ms to 6.55 ms by 0.41 ms step approximately). Figure [6.6] is the plot of the pixel variances ($V_m/2$) against the mean pixel values (S_m) in open circles. The slope of the least-square fitted line (the gain of the control system in ADU/e⁻) is about 1.83. The square root of its y-intercept (the overall system noise in ADU) is about 25.

The mean pixel values in the sub-areas of the image frames acquired show a moderately linear response to the exposure time. However, the gain and the overall noise of the control system obtained from this experiment were concluded unreliable from three reasons. Firstly, the data points in Figure [6.6] were widely scattered along

the least-square fitted line. Secondly, because the gain of the signal processing chain was lowered significantly in this experiment to protect ADCs from high negative glitches observed on the oscilloscope, such a high system gain was not expected. Thirdly, the y-intercept exhibited some difference from the variance of the overscan regions, which is another measure used for its estimation.

Initially, taking the linear response of the control system into account, this result was believed to have been caused not by any serious hardware problem with the control system but by the relatively short exposure time or the low system gain. It is believed that the video signal was not amplified high enough to overcome the high readout noise floor that is inherent in any high-speed CCD control system. The characteristics of the LED light source were also conceived as the possible cause. Its slow time response could have caused the unreliable result in this experiment and its non-linear response to current flow could have caused the small deviation of data points from the least-square fitted line in Figure [6.4].

However, the analysis of the overscan regions of image frames used to obtain Figures [6.5] and [6.6] revealed that their mean values and variances exhibited some variation. Their mean values had the average of about 109 ADU with the deviation of about 3 and their variances were in the range of 37 to 60 approximately. There are three possible reasons. Firstly, the system hardware is unstable. Secondly, the reference and signal sampling of the video signal are made incorrectly. Thirdly, the system hardware is subject to some pattern noise. The observation of the image frames revealed the strong sinusoidal noise pattern that is also observed in Figure [6.7.a].

To uncover the possible cause or combination of causes of the unreliability, more experiments and the in-depth analysis of the system characteristics are required. A sufficient number of data points needs to be acquired with a stronger light source, longer integration times, or a higher system gain. To identify the cause of instability or the source of the strong noise, experiments with all inputs of signal processing chains grounded will be effective. In addition, as can be seen in Figure [6.7.a], the image frames acquired from the bottom right output node exhibit a noise pattern that is not observed in those acquired from the bottom left one. It implies that the video signals processed even on the same module board are coupled by different noise sources. This

phenomenon must be examined by conducting the mean-variance tests on video signals from each signal processing chain and their combinations. They will give the quantitative characteristics of each circuit and the amount of the cross-talk between them.

6.5 Suggestion for Future Work

As pointed out in Chapter 1, another critical attribute of a high-speed CCD control system for adaptive optics is the low readout noise. For a telescope with a given number of subapertures, fainter natural guide stars can be used for wavefront sensing if the CCD control system has higher SNR and is not shot-noise-limited. Therefore, the chance to find a reference star increases. Moreover, because the incoming light is divided for a number of subapertures, more subapertures can be used for wavefront sensing at shorter wavelengths with higher wavefront sampling frequency if the CCD control system has higher SNR. In addition, the wavefront measurement error is inversely proportional to the square of SNR in general[7]. Hence, a CCD control system with higher SNR can give more accurate wavefront measurement. It is very significant that the ACCD designed by Levine *et al.*[64] employs skipper amplifiers that enable the non-destructive multiple sampling of the signal charges at its output nodes. As the result, high SNR is achieved because the noise from the output transistor contributes only once for multiple sampling of the signal charges. However, because many WFS are designed to operate at relatively high frame rates in the order of a few KHz, its CCD control system is inherently subject to the high-frequency switching noise. In order to increase the CCD readout time and thus to reduce the high-frequency switching noise, most WFS's employ multiport CCDs. In the previous section, several reasons are suggested for the failure of the quantitative analysis of the control system described in the work of this thesis. To be of real use as a part of a WFS for adaptive optics, it needs some optimisation and improvement in terms of the readout noise after more testing.

The inherent high-frequency switching noise can be reduced most effectively by putting clock drivers in the proximity of the CCD. It shortens the length of wires between clock drivers and CCD clock gates and reduce the generation and coupling of

noises. In this work, clock drivers are integrated on two module boards to reflect the design idea to build a versatile control system. The control system was tested with two Signal Processing Boards placed next to the CCD Board. The switching noise is believed to decrease by putting two Clock Boards next to the CCD Board and using shorter coaxial cables for the clock pulses. Most components on the prototype module boards were wire-wrapped. Some definite spacing between module boards on the system sub-rack was necessary because of the length of wire-wrapping pins. The most effective way to reduce the switching noise is to use carefully designed PCBs for all hardware modules. Separate ground planes must be used for components that generate the high-frequency switching noise. Power supply planes must be also implemented. The PCB tracks for switching signals must be short and close to the ground plane. The signal tracks running in parallel must be also avoided to reduce the noise coupling.

The sequencer logic is designed to incorporate the Bitmap and static DAC control methods and to provide the user with the facility to specify the number of transfers. Being programmed in an EEPLD, EPM7256E-12, it consumes less power and occupies less physical space than those based discrete devices. In addition, its operation is more reliable and predictable. However, as pointed out earlier, it is less flexible than a sequencer based on a microprocessor or a DSP in performing sophisticated readout functions. For example, a fairly complicated combinational and sequential logic is needed to perform the readout of a region of interest, especially for multiport CCDs. It may be a good exercise to programme the sequencer logic that can perform these functions into several FPGAs. A hybrid sequencer that consists of a processor and an FPGA is also promising. The FPGA can perform the sequential logic functions that require high-speed operation such as the generation of clock pulses whilst the processor performs other logic functions such as the control of number of transfers. The overhead time between transfers can be minimised by using high-speed processors recently developed. Many new FPGAs with higher logic densities and operational speeds such as those in the MAX 9000 family from Altera are also available. Their use can increase the readout speed of the control system. It must be noted, however, that every sequencer with the bit-slicing control method needs to allocate a specific number of clock cycles to each transfer. For example, even with an

FPGA that operates at 150 MHz, the pixel sampling frequency is about 18.8 MHz when 8 clock cycles are allocated to the horizontal transfer.

Another critical segment that governs the performance of the control system in terms of the readout speed is the clock driver. In this work, much endeavour was put into the construction of high-speed clock drivers that generate clock pulses with the voltage swings defined by octal DACs. The clock drivers built with MOSFET drivers, EL7212 and DS0026, meet the required frame rate for wavefront sensing. Although it is advantageous to drive DS0026's with PNP transistors rather than external capacitors, they were found inadequate for horizontal clock gates due to their long turn-off time. The use of transistors with higher switching speeds must be considered in case carefully designed PCBs are used for hardware module boards in the future. Being superior to DS0026, the EL7212 was not employed as the clock driver because it caused frequent damage to other components. It will be beneficial to experiment with the possibility to drive it either with transistors or with external capacitors. Nevertheless, their maximum switching frequency in the order of a few MHz and incapability to support input pulses with low duty cycles limits their use for high-speed CCD control systems. The clock driver similar to the one by Gradl [36] that is made of a pair of emitter-coupled high-speed transistors or a CCD integrated with clock drivers will have to be taken into consideration.

It was observed that several image frames acquired in the continuous readout mode were frequently corrupted. Originally, the Image RAMs are employed to store image frames only in the burst readout mode. In the continuous readout mode, new image frames are overwritten on old ones. Even though the Image RAMs are addressed to store a finite number of image frames, it is believed that there is a timing problem either in storing or retrieving them. For the continuous readout of the CCD to compensate the distorted wavefront in real time, a different way to handle the image frames is needed. If all 64×64 pixels of the Loral frame-transfer CCD are read out in 1 ms (1000 Hz) and sampled by 12-bit ADCs, the raw output data rate is approximately 6.14 Mbytes/sec. Because the wavefront sensor does not require all image frames to be stored, the use of 'ping-pong' image buffers or FIFO(first-in-first-out) buffers seems practical. For other applications that require higher frame rates, the handling of the large amount of image data must be considered with attention.

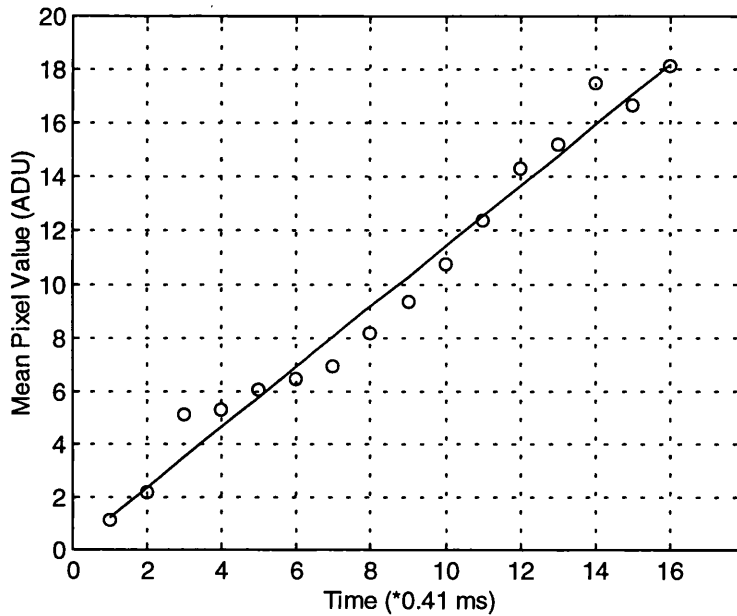


Figure [6.5] Response of the Loral CCD and the control system to different exposure time that was controlled by the frame-transfer Bitmap
The open circles are the measured data points and the solid line is the least-square fitted line of them.

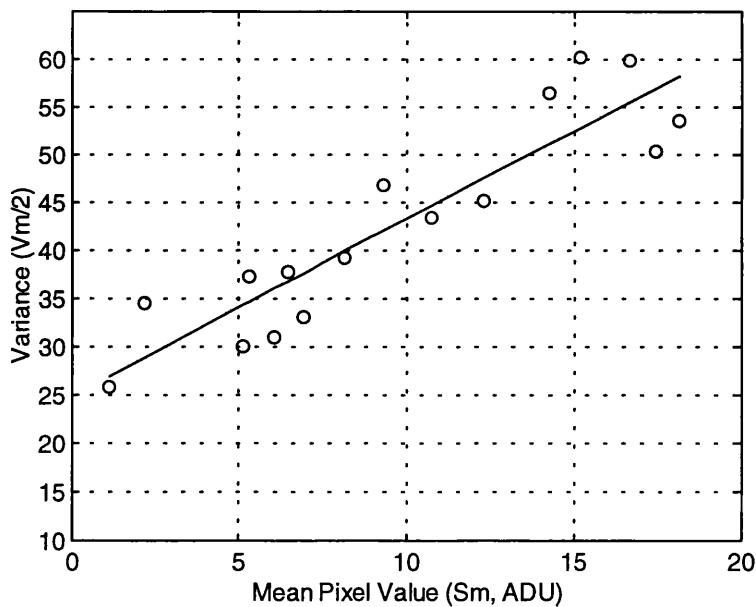
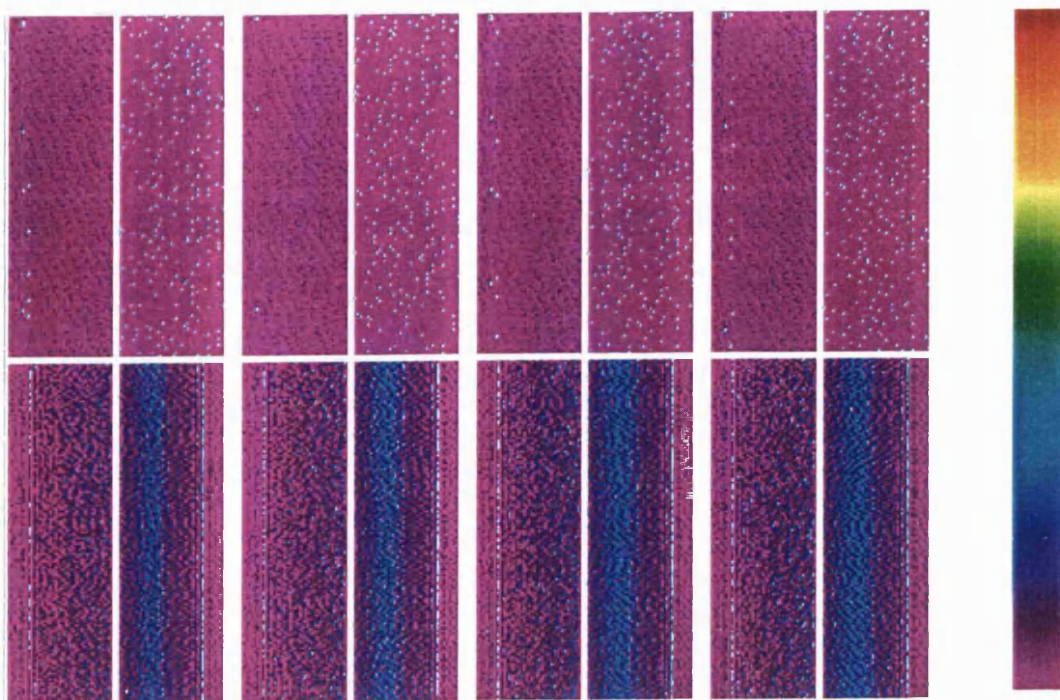
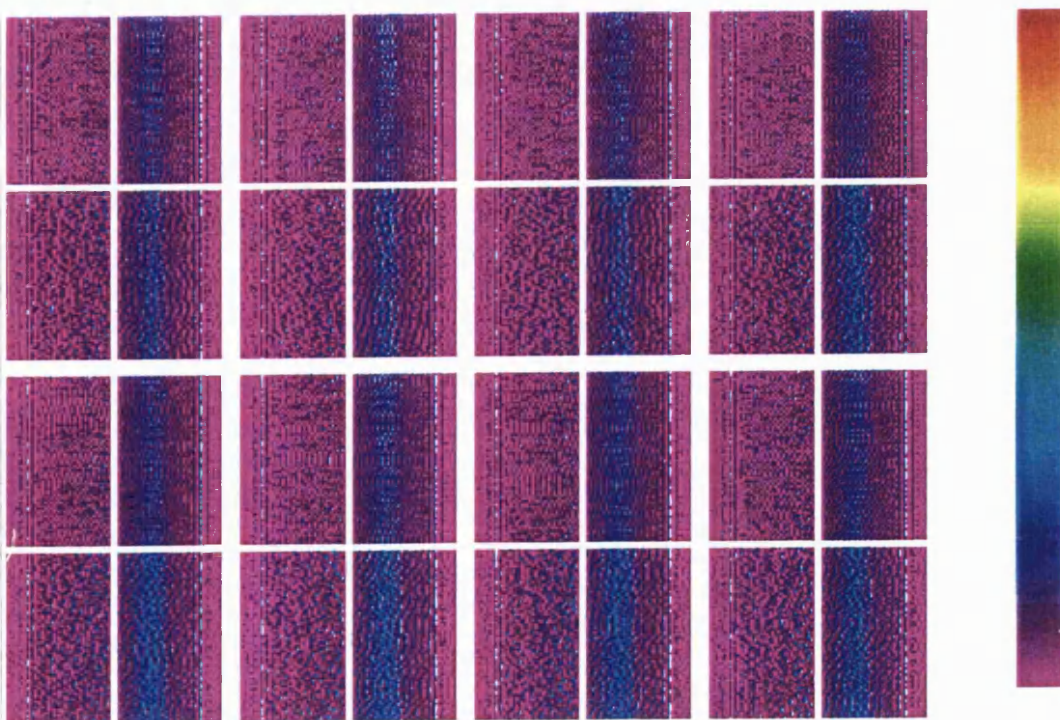


Figure [6.6] Plot of the pixel variances against the mean pixel values for the data points in Figure [6.5]. The open circles are the data points and the solid line is the least-square fitted line of them. The slope of the line is about 1.83 and its y-intercept is about 25.

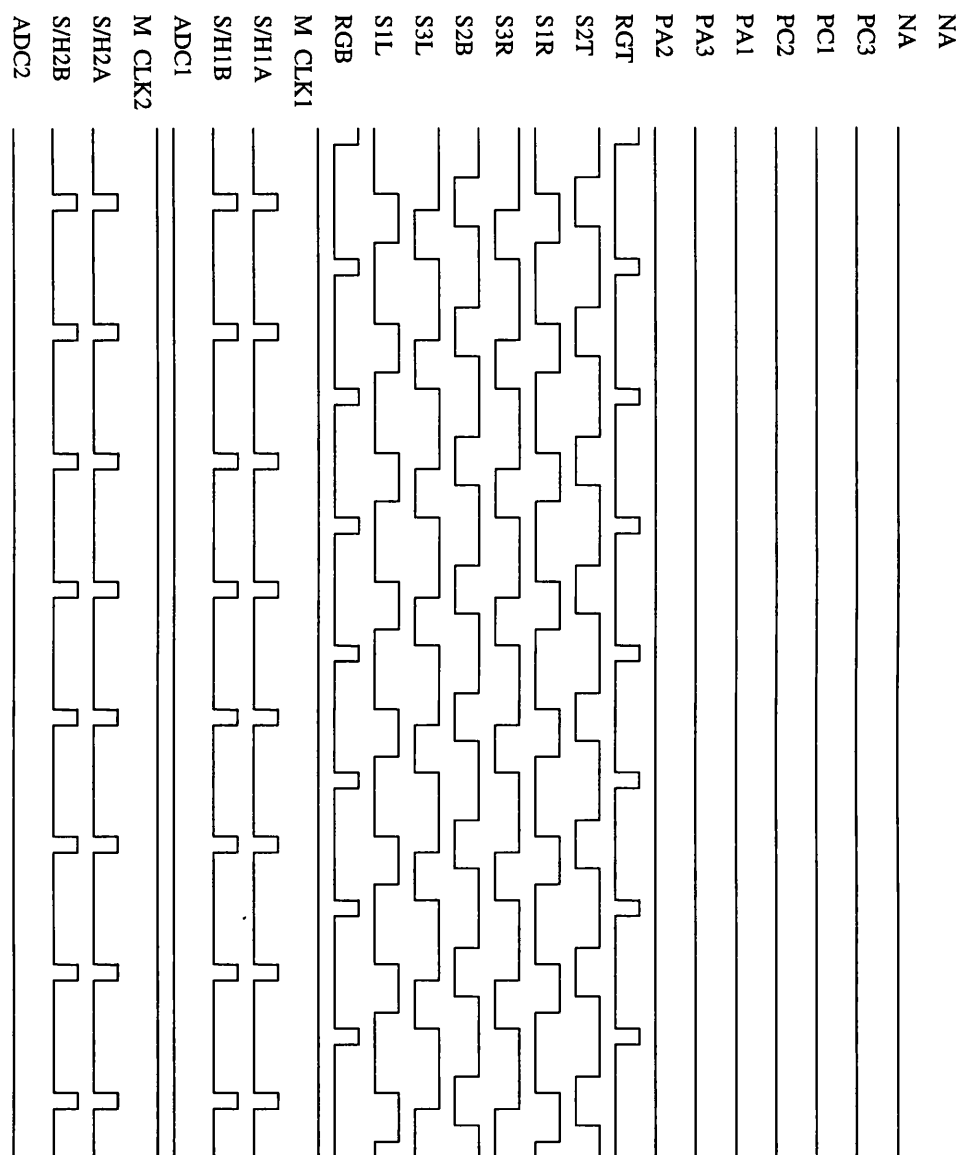


a. Four consecutive image frames acquired through bottom two output nodes



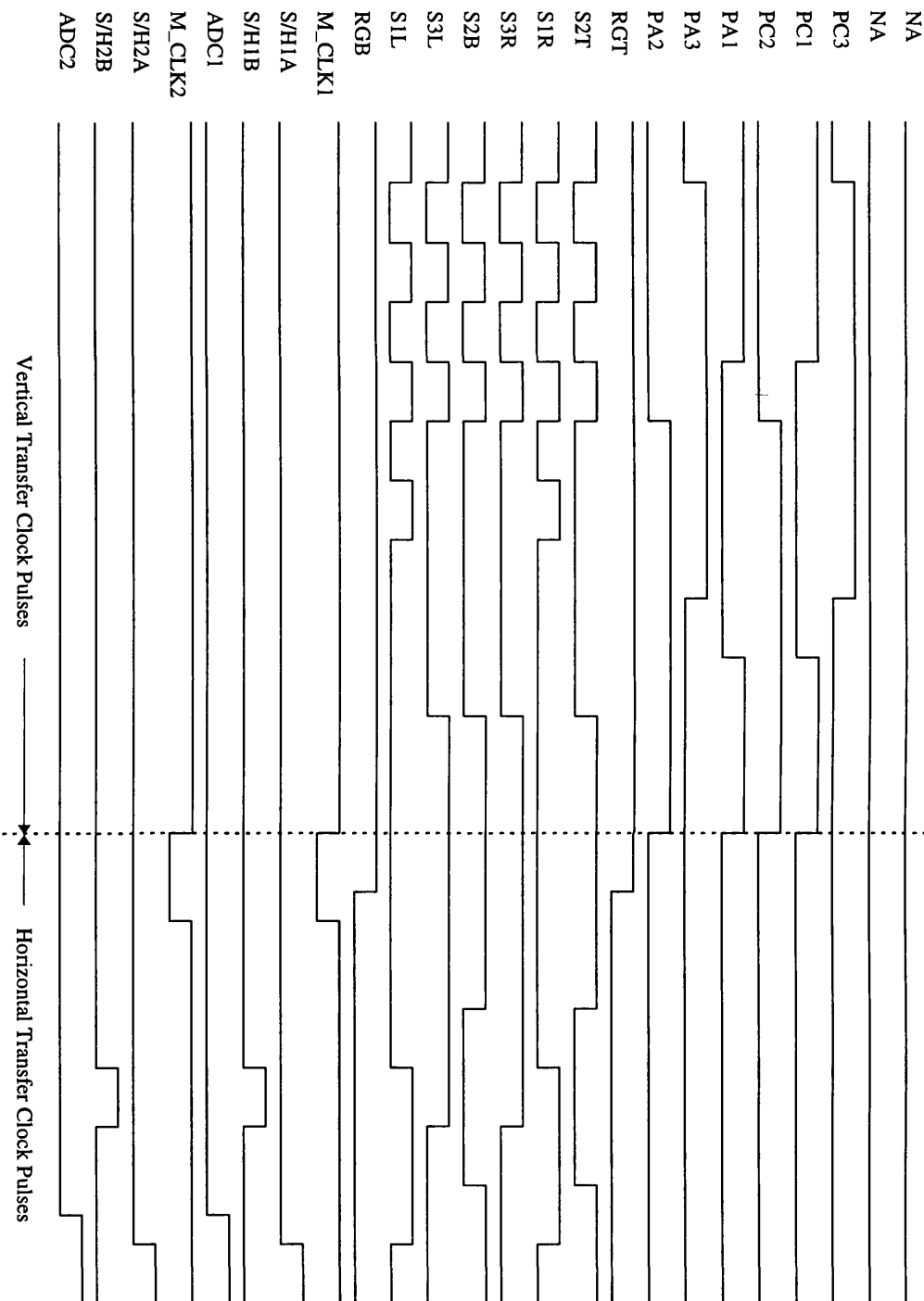
b. Eight consecutive image frames acquired through all output nodes

Figure [6.7] Two sequences of image frames acquired from the Loral CCD by the control system in the burst readout mode. The vertical palette bar is also shown at the right hand side. Its bottom colour is for 0 ADU and the top colour is for 255 ADU.



a. Clock pulse streams to transfer the signal charges in a single line of pixels over the frame-transfer period. The top Bitmap channel was used to supply the on/off pulse to the LED light source. The S/H1A and S/H1B Bitmap channels show that the video signal is sampled simultaneously by two S/H amplifiers to prevent the discharge of their internal capacitors over the frame-transfer period.

Figure [6.8] Streams of clock pulses for the frame, vertical, and horizontal transfers used in the mean-variance test (continued over)



b. Clock pulse streams in front of the vertical dotted line are for the transfer of the signal charges in a single line of pixels in storage sections. They show the clocking of horizontal gates at the beginning to prevent the discharge of the external capacitors to the inputs of MOSFET drivers. The clock pulse streams after the dotted line are for the transfer of the signal charges in a pixel in horizontal registers.

Figure [6.8] Streams of clock pulses for the frame, vertical, and horizontal transfers used in the mean-variance test

Conclusion

The performance of a large ground-based astronomical telescope is severely limited by the air turbulence in the Earth's atmosphere. Most telescopes under construction or design incorporate adaptive optics systems in order to correct the distorted wavefront and thus to obtain diffraction-limited images. The WFS is one of the fundamental elements in an adaptive optics system that is used to sample the distorted wavefront. It needs to sample the wavefront with enough spatial resolution and temporal sampling frequency for wavefront correction in real-time. It also needs to have high sensitivity to work on faint objects with many subapertures. The CCD has superior physical characteristics to other means of optical detection and it is the most widely used for WFS's.

This thesis deals with the design and construction of the high-speed CCD control system for adaptive optics systems. The design goals for the control system are the readout frequency of approximately 1000 Hz for a CCD of 100×100 pixels, a versatile system, a modular and compact system, and a user-friendly interface. The features of the control system are summarised as follows.

- *Versatility* : The control system incorporates the Bitmap and static DAC control methods. In the Bitmap control method, the various streams of CCD clock timing pulses are represented with simple streams of digital numbers. The streams of control timing pulses can be easily integrated into the Bitmaps. The Bitmaps are converted to the streams of CCD clock pulses with the static DAC control method. In this method, DACs define the high and low voltage levels of clock pulses, the control of which is

accomplished by the user through the control software. The definition of the bias voltage levels is also made with DACs. In addition, the user is allowed to specify the numbers of frame, vertical, and horizontal charge transfers in accordance with the physical format of the CCD.

- *Readout frequency* : The control system employs the high-speed sequencer and CCD clock drivers. Physically, the sequencer consists of a high-speed EEPLD, EPM7256E-12 and three high-speed SRAMs (Bitmap RAMs) that operate at 80 MHz. The Bitmap control method is embodied in the sequencer logic. The sequencer logic is also designed to ensure the reliable and predictable provision of the streams of Bitmaps in high speed. The clock driver is built with the high-speed MOSFET driver that has the capability to supply high currents to capacitive loads. The clock driver can sample the signal charges at the maximum frequency of approximately 10 MHz depending on its load capacitance.

- *Modularity and compactness* : The prototype control hardware is constructed based on the Loral 64×64 frame-transfer CCD that has four output nodes. It consists of various modules that are defined according to their distinctive functionality; host PC, main power supply unit, Voltage Regulation Board, Bitmap Generation Board, Bias Board, two Clock Boards, two Signal Processing Boards, and CCD Board. With the clear conceptual clarification between the classes through the inheritance and dynamic binding, the GUI and control software running on the host PC is also structured and modular. As a whole, the control system is compact, modular, portable, and stand-alone.

- *User-interface* : The GUI establishes an effective and easy-to-use graphical environment to control numerous system parameters provided by the system hardware. With the Bitmap Editor in the GUI, a user can define and modify the streams of Bitmaps precisely. The GUI provides the graphical tool to manage various DACs that are used to define the voltage levels of CCD biases and clock pulses. The control of other parameters such as the readout mode and the integration time are performed in the similar graphical environment. The GUI also includes the control software to perform the writing of system parameters and Bitmaps, the readout of the CCD, and the display of the acquired image frames on the host PC monitor.

The control system features the versatility through which a user can control various system parameters. Therefore, the optimum performance with a given CCD can be achieved without any physical modification. It can also drive virtually any frame-transfer CCD of up to 1024×1024 pixels in addition to full frame CCDs when the frame-transfer Bitmap is kept to minimal or is used for light integration. The control system is a modular, compact, and stand-alone system, which ensures the rapid evaluation and modification. The GUI provides an easy-to-use graphical control environment for a user to maximise the versatility of the control system.

The prototype hardware module boards were constructed and integrated on a sub-rack. The complete control system consists of the main power supply unit, the sub-rack, and the host PC as shown in Figure [6.3]. The overall control of the system is performed through the GUI and control software running on the host PC. Image frames were successfully acquired from different combinations of output nodes of the Loral CCD at the maximum pixel sampling frequency of 10 MHz. Figure [6.7] shows two sequences of image frames acquired at the 5 MHz pixel sampling frequency in the burst readout mode. The linear response of the control system to different levels of the light intensity was also verified as shown in Figure [6.4]. However, the mean-variance test performed to analysis quantitatively the characteristics of the control system failed to produce a reliable result. Several reasons for the failure were suggested in subsection 6.4.3.

For a 10-m telescope with a Shack-Hartmann type sensor and the seeing conditions in Table [3.1], at least a 256×256 CCD is needed to meet required the spatial resolution for all bands when four pixels are assigned to each subaperture. The control system can be easily configured to drive the 256×256 CCD and meets the required sampling frequency for wavefront correction in all bands when the CCD is read out through four output nodes in parallel. If 4×4 neighbouring pixels are assigned to each subaperture in order to obtain higher position accuracy or to reduce the cross-talk, a 512×512 CCD is needed for wavefront correction in all bands. The control system fails to produce the required sampling frequency with four output nodes running in parallel. In this case, the use of a CCD with more output nodes is desirable.

The high-speed CCD control system is believed to eliminate the limitation imposed on the current WFS developed at OSL in terms of the readout speed. It also proved the effectiveness of the Bitmap and static DAC control methods elaborated at OSL for high-speed control systems. However, the control system needs more verification and optimisation in addition to the precise quantitative analysis of its characteristics. To work on faint objects with a number of subapertures, the readout noise, especially the high-frequency switching noise, must be significantly reduced. The versatility of the control system also promises its usefulness in other scientific and engineering fields; the testing of the large optics, motion analysis and high-speed image acquisition, and x-ray synchrotron radiation research.

Bibliography

- [1] *1996-1997 Databook*, 1996/1997, Dalsa Inc.
- [2] Abbott, T. and Opal, C., 1987, *A high speed CCD photometer for McDonald Observatory*, in *Instrumentation for Ground-based Optical Astronomy*, Proc. 9th Santa Cruz Summer Workshop in Astronomy and Astrophysics, p. 386, Lick Observatory
- [3] Acton, P., Agnew, G., Cotton, R., Hedges, S., McKemey, A., Robbins, M., Roy, T., Watts, S., Damerell, C., English, R., Gillman, A., Lintern, A., Su, D., and Wickens, F., 1991, *Future potential of charge coupled devices as detectors of ionising radiation*, Nucl. Instr. Meth. Phys. Res., A 305, p. 504
- [4] Babey, S. K., Anger, C. D., and Green, B. D., 1985, *Digital charge coupled device camera system architecture*, Proc. SPIE, 570, p. 39
- [5] Babcock, H. W., 1953, *The possibility of compensating astronomical seeing*, Proc. Astron. Soc. Pac., 65, p. 229
- [6] Beale, F. and Leach, R. W., 1990, *Video processor and clock driver circuitry for CCD cameras*, in Proc. of the CCDs in Astronomy Conference, Astr. Soc. of Pacific Conference Series, no. 8, ed. Jacoby, G. H., p. 180
- [7] Beckers, J. M., 1993, *Adaptive Optics for Astronomy : Principles, Performance, and Applications*, Annu. Rev. Astron. Astrophys., 31, p. 13
- [8] Beynon, J. D. E. and Lamb, D. R.(eds.), 1980, *Charge-coupled devices and their applications*, McGraw-Hill Book Comp. Ltd.
- [9] Bigelow, B. C., 1996, *PhD Thesis*, University of London
- [10] Bigelow, B. C., Walker, D. D., and Bingham, R. G., 1994, *Optimising a deformable secondary mirror for adaptive optics*, Proc. SPIE, 2201, p. 800
- [11] Bigelow, B. C., Wilson, R. W., Jenkins, C. R., and Walker, D. D., 1995, *Analysis, simulation and performance comparison of adaptive optical correction with deformable secondary mirrors*, Report to UK Adaptive Optics Programme, Durham, UK
- [12] Bortoletto, F. and D'Alessandro, M., 1986, *Universal CCD-controller system*,

- Rev. Sci. Instr., vol. 57, no. 2, p. 253
- [13] Boyle, W. S. and Smith, G. E., 1970, *Charge Coupled Semiconductor Devices*, Bell System Tech. J., vol. 49, no. 4, p. 587
- [14] Boyle, W. S. and Smith, G. E., 1970, *Experimental Verification of the Charge Coupled Device Concept*, Bell System Tech. J., vol. 49, no. 4, p. 593
- [15] Bredthauer, R. A., 1985, *Advances in CCD Technology*, Proc. SPIE, 570, p. 35
- [16] Bredthauer, R. A., Chandler, C. E., Janesick, J. R., McCurnin, T. W., and Sims, G. R., 1987, *Recent CCD Technology Developments*, Instrumentation for Ground-based Optical Astronomy, Proc. 9th Santa Cruz Summer Workshop in Astronomy and Astrophysics, p. 486, Lick Observatory
- [17] Bregman, J. D. and Doorduyn, A., 1986, *New design concepts for compact CCD controllers*, Proc. SPIE, 627, p. 616
- [18] Burrows, D. N., Berthiaume, G. D., Catalano, M. A., Entingh, T. J., Garmire, G. P., Larkin, C., Nousek, J. A., and Weaver, G. M., 1990, *A low-cost, versatile CCD controller*, in Proc. of the CCDs in Astronomy, Astr. Soc. of Pacific Conference Series, no. 8, ed. Jacoby, G. H., p. 165
- [19] Burt, D. J., 1991, *CCD performance limitations: theory and practice*, Nucl. Instr. Meth. Phys. Res., A 305, p. 564
- [20] *CCD Databook*, 1994/1995, Loral Fairchild
- [21] *CCD Products*, 1992/1993, Thomson-CFS
- [22] Castelli, C., Wells, A., McCarthy, K., and Holland, A., 1991, *Soft X-ray response of charge coupled devices*, Nucl. Instr. Meth. Phys. Res., A 310, p. 240
- [23] Clampin, M., Durrance, S. T., Golimowski, D. A., Barkhouser, R., Westphal, J. A., Heidtmann, D. L., and Blouke, M. M., 1990, *High speed quadrant CCDs for adaptive optics*, in Proc. of the CCDs in Astronomy, Astr. Soc. of Pacific Conference Series, no. 8, ed. Jacoby, G. H., p. 367
- [24] Clarke, R., 1994, *CCD X-ray detectors: opportunities and challenges*, Nucl. Instr. Meth. Phys. Res., A 347, p. 529
- [25] Collect, M. G., 1985, *Solid State Image Sensors*, Proc. SPIE, 591, p. 82

- [26] Conder, A. D., Dunn, J., and Young B., 1995, *Miniature, vacuum compatible 1024×1024 charge-coupled device camera for x-ray, ultraviolet, or optical imaging*, Rev. Sci. Inst., vol. 66, no. 1, p. 709
- [27] *Data Book*, 1993, Altera Co.
- [28] Doel, P., 1995, *A comparison of Shack-Hartmann and curvature sensing for large telescopes*, Proc. SPIE, 2534, p. 265
- [29] Doty, J. P., Luppino, G. A., and Ricker, G. R., 1987, *Design of low noise, high performance x-ray charge-coupled-device cameras*, Opt. Eng., vol. 26, no. 10, p. 1055
- [30] Etoh, T. and Takehera, K., 1995, *Needs, requirements and new proposals for ultra-high-speed videocameras in Japan*, Proc. SPIE, 2513, p. 231
- [31] Faison, T., 1991, *Graphical User Interfaces with Turbo C++*, SAMS
- [32] Fiorucci, G., Bourquin, J.-P., Bovet, D., Bovet, E., Egger, J.-P., Heche, C., Nussbaum, C., Schenker, D., Variel, D., and Vuilleumier, J.-M., 1990, *CCDs As Low-Energy X-ray Detectors*, Nucl. Instr. Meth. Phys. Res., A 292, p. 141
- [33] Gemini Newsletter, *Science Drives for Adaptive Optics on Gemini North*, Jun 1996
- [34] Gentles, B., Wells, M., Longmore, A., and Myers, R., 1995, *Natural guide star AO system for the William Herschel telescope*, Proc. SPIE, 2534, p. 62
- [35] Gleckler, A. D. and Wizinowich, P. L., 1995, *W. M. Keck Observatory adaptive optics program*, Proc. SPIE, 2534, p. 386
- [36] Gradl, D. A., 1981, *250 Mhz CCD Driver*, IEEE J. Solid-State Circuits, vol. 16, no. 2, p. 100
- [37] Gray, P. M., Lloyd-Hart, M., Angel, J., McCarthy, Jr., D., Sandler, D., Martinez, T., Close, L., Brusa, G., Bruns, D., McLeod, B., Ryan, P., Groesbeck, T., Wittman, D., Jacobsen, B., Hughes, J., Hunten, M., and Cheselka, M., 1995, *FASTTRAC II near-IR adaptive optics system for the Multiple Mirror Telescope: description & preliminary results*, Proc. SPIE, 2534, p. 2
- [38] Greenwood, D. P., 1977, *Bandwidth specification for adaptive optics system*, J. Opt. Soc. Am., vol. 67, no. 3, p. 390

-
- [39] Gunn, J. E. and Westphal, J. A., 1981, *Care, feeding, and use of charge-coupled device imagers at Palomar Observatory*, Proc. SPIE, 290, p. 16
- [40] Gursky, H., 1981, *Impact of solid-state imaging on optical astronomy*, Proc. SPIE, 290, p. 2
- [41] Han, W., 1993, *PhD Thesis*, University of London
- [42] Hardy, J. W., 1978, *Active Optics : A New Technology for the Control of Light*, Proc. IEEE, vol. 66, no. 6, p. 651
- [43] Hayes, R. and Heidtmann, D. L., 1987, *Dual-channel charge-coupled device for high speed signal acquisition*, Opt. Eng., vol. 26, no. 9, p. 829
- [44] Higgins, T. V., *The technology of image capture*, Laser Focus World, Dec 1994, p. 53
- [45] Hiller, W. J. and Kowalewski, T. A., 1989, *Application of the frame-transfer charge-coupled device for high speed imaging*, Opt. Eng., vol. 28, no. 2, p. 197
- [46] Hori, T., 1995, *Progressive scan interline CCD camera*, Proc. SPIE, 2416, p. 17
- [47] Hubin, N. and Noethe, L., 1993, *What is Adaptive Optics?*, Science, vol. 262, p. 1345
- [48] Hughes, G., *High frame rate electronic imaging*, Optics & Photonics News, Apr 1995, p.21
- [49] *Image Sensing and Solid State Camera Products*, 1995/1996, EG&G Reticon
- [50] Janesick, J. R., *Focus on high-speed scientific CCDs*, Nov 1995, OE Reports, SPIE
- [51] Janesick, J. R. and Blouke, M., *Past, Present, & Future*, Optics & Photonics News, Apr 1995, p. 17
- [52] Janesick, J. R. and Elliott, S. T., 1991, *History and Advancements of Large Area Array Scientific CCD Imagers*, Astronomical Society of Pacific Conference Series, Tucson, AZ, USA
- [53] Janesick, J. R., Elliot, T., Collins, S., Blouke, M. M., and Freeman, J., 1987,

Scientific charge-coupled devices, Opt. Eng., vol. 26, no. 8, p. 692

- [54] Janesick, J. R., Elliot, T., Daud, T., McCarthy, J., and Blouke, M., 1985, *Backside charging of the CCD*, Proc. SPIE, 570, p. 46
- [55] Jorden, P. R., 1990, *CCDs for 1990s*, in *New Windows to the Universe*, vol. 2, ed. Sanchez, F. and Vazquez, M., p. 465, Cambridge University Press
- [56] Jorden, P. R., 1990, *The use of CCDs on optical telescopes*, in *Modern Technology and its Influence on Astronomy*, ed. Wall, J. V. and Boksenberg, A., p. 271, Cambridge University Press
- [57] Jorden, P., *CCDs Past, Present and Future*, Newsletter of RGO, Mar 1990, p. 20
- [58] Kerr III, J., Clark, C., and Smith, S., 1994, *CFHT's generation III controller: a multi-amplifier FPA readout system*, Proc. SPIE, 2198, p. 980
- [59] Kim, Y.-S., 1995, private communications
- [60] Lawler, W., Harrison, L., and Aciu, A., 1994, *High-frame-rate image acquisition system*, Proc. SPIE, 2273, p. 38
- [61] Leach, R. W. and Beale, F. L., 1990, *Design and operation of a multiple readout CCD camera controller*, Proc. SPIE, 1235, p. 284
- [62] Leach, R. W. and Denune, J., 1994, *Noise and readout performance of a multiple readout CCD controller*, Proc. SPIE, 2198, p. 821
- [63] Lee, R. and Tyson, J. A., 1987, *A large area CCD imaging system*, Instrumentation for Ground-based Optical Astronomy, Proc. 9th Santa Cruz Summer Workshop in Astronomy and Astrophysics, p. 742, Lick Observatory
- [64] Levine, B. M., Janesick, J. R., and Shelton, J. C., 1994, *Development of a low noise high frame rate CCD for adaptive optics*, Proc. SPIE, 2201, p. 596
- [65] Levine, P. A., 1992, *High frame rate multi-readout CCD imager and camera*, Proc. SPIE, 1757, p. 257
- [66] Lippman, S. B., 1991, *C++ Primer*, 2nd ed., Addison-Wesley Pub. Comp.
- [67] Lloyd-Hart, M., Angel, J., Sandler, D., Groesbeck, T., Martinez, T., and

- Jacobsen, B., 1996, *Design of the 6.5 m MMT adaptive optics system, and results from its prototype system FASTTRAC II*, Proc. SPIE, 2871
- [68] Loh, E. D., 1981, *Princeton observing and picture-processing equipment*, Proc. SPIE, 290, p. 150
- [69] Lula, J. J., Miethig, M., Litwiller, D. J., Doody, B. C., Washkurak, S. G., and Chamberlain, S. G., 1994, *A 23,000 frame per second 256×128 camera with digital control*, Proc. SPIE, 2173, p. 207
- [70] Lumb, D. H. and Nousek, J. A., 1992, *Energy and Time Response of CCD X-Ray Detectors*, IEEE Tran. Nucl. Sci., vol. 39, no. 5, p. 1379
- [71] Luppino, G. A., Bredthauer, R. A., and Geary, J. C., 1994, *Design of an 8192 x 8192 pixel CCD mosaic*, Proc. SPIE, 2198, p. 810
- [72] Luppino, G. A., Ceglio, N. M., Doty, J. P., Ricker, G. R., and Vallergera, J. V., 1987, *Imaging and nondispersive spectroscopy of soft x rays using a laboratory x-ray charge-coupled-device system*, Opt. Eng., vol. 26, no. 10, p. 1048
- [73] MacKay, C. D., 1994, *Reports on Tests of Loral 64 x 64 Frame Transfer CCD*, Institute of Astronomy, University of Cambridge
- [74] MacKay, C. D., 1996, private communications
- [75] McLean, I. S., 1989, *Electronic and Computer-Aided Astronomy*, Ellis-Horwood Ltd.
- [76] McNutt, M. J., Mattson, R. B., and Vu, A. N., 1990, *Schottky-barrier infrared focal plane array with novel readout structure*, IEEE J. Solid-state Circuits, vol. 25, no. 2, p. 602
- [77] Meyer, H. J., and Doherty, P. E., 1994, *Operational considerations for high frame rate CCD readout*, Proc. SPIE, 2173, p. 97
- [78] Meyers, S., 1992, *Effective C++*, Addison-Wesley Pub. Comp.
- [79] Millman, J. and Grabel, A., 1987, *Microelectronic*, 2nd ed., McGraw-Hill Book Comp. Ltd.
- [80] Mooney, J., *Charge-coupled devices in infrared imaging*, Optics & Photonics News, Apr 1995, p. 29

- [81] Myers, R., Longmore, A., Humphreys, R., Gilmore, G., Gentles, B., Wells, M., and Wilson, R., 1995, *The UK Adaptive Optics Programme*, Proc. SPIE, 2534, p. 48
- [82] Nicholls, T. W., Solomon, C. J., Wells, N., Wooder, N. J., and Dainty, J. C., 1994, *Experiments on Wavefront Sensing at La Palma*, Proc. SPIE, 2201, p. 555
- [83] Olivier, S., An, J., Avicola, K., Bissinger, H., Brase, J., Friedman, H., Gavel, D., Max, C., Salmom, J., and Waltjen, K., 1995, *Performance of laser guide star adaptive optics at Lick Observatory*, Proc. SPIE, 2534, p. 26
- [84] Ott, H. W., 1988, *Noise Reduction Techniques in Electronic Systems*, 2nd ed., John Wiley & Sons, Inc.
- [85] O'Byrne, J., Bryant, J., Minard, R., Fekete, P., and Cram, L., 1995, *Adaptive Optics at the Anglo-Australian Telescope*, Pub. Astron. Soc. Australia, vol. 12, p. 106
- [86] O'Byrne, J., *Sharper Eyes on the Sky*, Sky & Telescope, Dec 1996, p. 20
- [87] Parker, D., *Optical detectors: research to reality*, Physics World, Mar 1990, p. 52
- [88] Pennington, T. L., Welsh, B. M., Roggemann, M. C., 1994, *Performance comparison of the shearing interferometer and Hartmann wavefront sensors*, Proc. SPIE, 2201, p. 508
- [89] Pennington, T., Swindle, D., Olier, M., Ellerbroek, B., and Spinhrine, J., 1995, *Performance measurements of Generation III wavefront sensors at the Starfire Optical Range*, Proc. SPIE, 2534, p. 327
- [90] Prokop, M., McCurnin, T. W., and Stradling, G., 1992, *Millisecond readout CCD camera*, Proc. SPIE, 1757, p. 268
- [91] Pugh, W. N., Lobb, D., Walker, D., and Williams, T., 1995, *Pupil-imaging wavefront gradient sensor*, Proc. SPIE, 2534, p. 312
- [92] Racca, R. G., Stephenson, O., and Clements, R. M., 1992, *High-speed video analysis system using multiple shutter charge-coupled device imagers and digital storage*, Opt. Eng., vol. 31, no. 6, p. 1369

- [93] Rigaut, F., Lai, O., Rouan, D., Salmon, D., Arsenault, R., Thomas, J., Veran, J. P., Gigan, P., Crampton, D., Fletcher, J. M., Stillburn, J., Boyer, C., and Jagourel, P., *Performance of the Canada-France-Hawaii Telescope Adaptive Optics Bonnette*, submitted to PASP
- [94] Robertson, D. and Mountain, M., 1994, *Gemini Instrumentation*, Proc. SPIE, 2198, p. 143
- [95] Robinson, L. B., Stover, R. J., Osborne, J., Miller, J. S., Vogt, S. S., and Allen, S. L., 1987, *Lick Observatory charge-coupled-device data acquisition system*, Opt. Eng., vol. 26, no. 8, p. 795
- [96] Rodricks, B. and Brizard, C., 1992, *Programmable CCD imaging system for synchrotron radiation studies*, Nucl. Instr. Meth. Phys. Res., A 311, p. 613
- [97] Rodricks, B., Huang, Q., Hopf, R., and Wang, K., 1994, *A large area detector for X-ray applications*, Nucl. Instr. Meth. Phys. Res., A 348, p. 572
- [98] Rose, A. and Weimer, P. K., *Physical limits to the performance of imaging systems*, Physics Today, Sep 1989, p. 24
- [99] Rousset, G., 1994, *Wavefront sensing*, in Adaptive Optics for Astronomy, ed. Alloin, D. M. and Mariotti, J.-M., p. 115, Kluwer Academic Publishers
- [100] Sandler, D., Lloyd-Hart, M., Martinez, T., Gray, P., Angel, R., Barrett, T., Bruns, D., and Stahl, S., 1995, *The 6.5 m MMT infrared adaptive optics system: detailed design and progress report*, Proc. SPIE, 2534, p. 372
- [101] Scribner, D. A., Kruer, M. R., and Killiany, J. M., 1991, *Infrared focal plane array technology*, Proc. IEEE, vol. 79, no. 1, p. 66
- [102] Shelton, J. C., Schneider, T., McKenna, D., and Baliunas, S. L., 1995, *First tests of the Cassegrain adaptive optics system of the Mount Wilson 100-inch telescope*, Proc. SPIE, 2534, p. 72
- [103] Simmons, J. E., Lillberg, J. W., McKee, R. J., Slice, R. W., Torrez, J. H., McCurnin, T. W., and Sanchez, P. G., 1994, *Megapixel imaging camera for expanded H α beam measurements*, Nucl. Instr. Meth. Phys. Res., A 305, p. 538
- [104] Smith, R. M., 1990, *CCD controller technology for 1990*, in Proc. of the CCDs in Astronomy Conference, Astr. Soc. of Pacific Conference Series, no. 8, ed. Jacoby, G. H., p. 153

- [105] Smutko, M., Chun, M., Shi, F., Scor, V., Wild, W., Larkin, J., and Kibblewhite, E., *An Overview of the Chicago Adaptive Optics System*, poster presented at the San Antonio meeting of the American Astronomical Society, Jan 1996
- [106] Spooren, R., 1994, *Standard charge-coupled device cameras for video speckle interferometry*, Opt. Eng., vol. 33, no. 3, p. 889
- [107] Stover, R. J., 1986, *High-speed charge-coupled device (CCD) imaging stellar photometer*, Proc. SPIE, 627, p. 195
- [108] Streetman, B. G., 1980, *Solid State Electronic Devices*, 2nd ed., Prentice-Hall, Inc.
- [109] Theuwissen, A. J. and Bosiers, J. T., *Consumer CCDs: Masterpieces of 3-D Integration*, Optics & Photonics News, Apr 1995, p. 32
- [110] Thompson, L. A. and Yao-Heng, X., 1995, *Laser beacon system for the UnISIS adaptive optics system at the Mt. Wilson 2.5 meter telescope*, Proc. SPIE, 2534, p. 38
- [111] Trauger, J. T., 1990, *Sensors for the Hubble Space Telescope Wide Field and Planetary Cameras (1 and 2)*, in Proc. of the CCDs in Astronomy Conference, Astr. Soc. of Pacific Conference Series, no. 8, ed. Jacoby, G. H., p. 217
- [112] Trimberger, S. M. (ed.), 1994, *Field-Programmable Gate Array Technology*, Kluwer Academic Publishers
- [113] Tsunemi, H., Kawai, S., and Hayashida, K., 1991, *Performance of the Charge-Coupled Device for Direct X-ray Detection in the Energy Range of 1-9 keV at the Synchrotron Radiation Facility*, Japanese J. of Appl. Phys., vol. 30, no. 6, p. 1299
- [114] Turko, B. T., 1985, *Fast readout of CCD images*, IEEE Tran. Nucl. Sci., vol. 32, no. 1, p. 576
- [115] Turner, L. K., Mantus, D. S., Ling, Y.-C., Bernius, M. T., and Morrison, G. H., 1989, *Development and characterization of a charge-coupled device detection system for ion microscopy*, Rev. Sci. Instr., vol. 60, no. 5, p. 886
- [116] Tyson, J. A., 1990, *Progress in low-light-level charge-coupled device imaging in astronomy*, J. Opt. Soc. Am., vol. 7, no. 7, p. 1231
- [117] Tyson, R. K., 1991, *Principles of Adaptive Optics*, Academic Press, Chapter 5

- [118] Weckler, G. P. and Tseng, H. F., 1985, *Evolution and Comparison of Silicon Image Sensors*, in IEE Photoelectronic Imaging Conference Publication, no. 253
- [119] Weckler, G. P., 1985, *The Solid State Image Sensor's Contribution To The Development of Silicon Technology*, Proc. SPIE, 570, p. 2
- [120] Weiss, S. A. and Wjeeler, S. V., *The CCD Camera: As Size and Cost Shrink, A \$2.1 Billion Industry Poises for New Growth*, Photonics Spectra, Jun 1994, p. 102
- [121] Wells, M., Nicholls, T., Wooder, N., Munro, I., and Dainty, C., 1995, *Infrared wavefronts on Mauna Kea*, Proc. SPIE, 2534, p. 258
- [122] Wen, D. D., 1974, *Design and Operation of a Floating Gate Amplifier*, IEEE J. Solid-State Circuits, vol. 9, no. 6, p. 410
- [123] West., J., Simpson, R. W., and Khoo, S., 1994, *A new camera for high speed, high-dynamic-range digital imaging at 4,000 frames/second (burst rate) and 14-16 bits/pixel*, Proc. SPIE, 2273, p. 2
- [124] Williams, T., 1991, *The circuit designer's companion*, Butterworth-Heinemann Ltd.
- [125] Yates, G. J., Pena, C. R., and Turko, B. T., 1994, *Characterization of multiport solid state imagers at megahertz data rates*, Proc. SPIE, 2273, p.13

Glossary

ACCD	Adaptive optics Charge-Coupled Device
ACK	Acknowledge
ADC	Analog-to-Digital Converter
ADU	Analog-to-Digital Converter Unit
APD	Avalanche Photodiode
BJT	Bipolar Junction Transistor
CCD	Charge-Coupled Device
CDS	Correlated Double Sampling
CID	Charge-Injection Device
CIM	Charge-Injection Matrix
CMOS	Complementary Metal-Oxide-Semiconductor
CTE	Charge Transfer Efficiency
DAC	Digital-to-Analog Converter
DN	Digital Number
DSP	Digital Signal Processor
EEPLD	Electrically-Erasable Programming Logic Device
EEPROM	Electrically-Erasable Programming Read-Only Memory
EPLD	Erasable Programming Logic Device
FIFO	First-In-First-Out
FPGA	Field-Programming Gate Array
ESO	European Southern Observatory
GUI	Graphical User-Interface
HST	Hubble Space Telescope
I/O	Input-and-Output
IBF	Input-Buffer-Full
IR	Infrared
LAB	Logic Array Block
LED	Light-Emitting Diode

LL	Lincoln Laboratory(Massachusetts Institute of Technology)
LLNL	Lawrence Livermore National Laboratory
LSB	Least Significant Bit
MAX	Multiple Array matrix
MIPS	Million Instructions Per Second
MIS	Metal-Insulator-Semiconductor
MMT	Multiple Mirror Telescope
MOS	Metal-Oxide-Semiconductor
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MPP	Multi-Pinned Phase
MSB	Most Significant Bit
NIR	Near-InfraRed
NMOS	N-type Metal-Oxide-Semiconductor
OBF	Output-Buffer-Full
OSL	Optical Science Laboratory
PC	Personal Computer
PCB	Printed Circuit Board
QE	Quantum Efficiency
QEH	Quantum Efficiency Hysteresis
RAM	Random Access Memory
S/H	Sample-and-Hold
SNR	Signal-to-Noise Ratio
SPIE	Society of Photo-optical Instrumentation Engineers
STB	Strobe
TTL	Transistor-Transistor Logic
UV	Ultra-Violet
WFPC	Wide Field and Planetary Camera
WFS	Wavefront Sensor
WHT	William Herschel Telescope
WWW	World Wide Web

Acknowledgements

Throughout this work, my parents and sisters showed the deepest affection towards me and gave the endless encouragement to me. Without them, this work would never have been possible. I would like to thank them sincerely ahead of anyone else.

Prof. S. D. Choi, the director of the Satellite Technology Research Center(SaTReC) in the Korea Advanced Institute of Science and Technology(KAIST), granted me the opportunity to pursue my Ph.D. at the Optical Science Laboratory(OSL) in the University College London(UCL). I would like to appreciate the confidence and belief that he had on me in spite of my poor ability.

Dr. D. D. Walker, my Ph.D. supervisor and the director of OSL, initiated this work and guided and encouraged me whenever I became astray. He also endured my frequent diversion from the work with tolerance and belief. I am greatly indebted to him. Dr. H. Jamshidi and Mr G. Nixon in OSL gave me the invaluable practical advice from their experience whenever I encountered unexpected hardware problems. The construction of the prototype system hardware would have been impossible without their assistance. They demonstrated to me the importance of experience to be a *real* engineer. Dr. H. Jamshidi also kindly revised some of my thesis and made corrections, which I would like to acknowledge.

Dr. W. Han, a former member of OSL, shared his experience on the CCD and control systems with me, established the foundation for the work of this thesis, and advised me at the beginning of the system design. In place of expressing my gratitude, I would like to recall those happy days we had together.

Dr. C. MacKay in the University of Cambridge kindly provided one of the Loral frame-transfer CCDs and some valuable information for the control system. I would like to express my gratitude to him. Mr Van Daalen in the Royal Holloway and New Bedford College allowed me to use the master programming unit for Altera EEPLD devices. I am indebted to his kindness.

I would like to give my special thanks to Dr. A. Radley and Mr W. N. Pugh, former members of OSL, who introduced the object-oriented programming language C++ to me and shared their experience and knowledge on the C++ programming. I feel

grateful to other OSL colleagues for their help during my Ph.D. years, especially Mr M. Dryburgh and Mr B. Mohan. I would like to appreciate the assistance of other members in the Dept. of Physics and Astronomy.

I have to admit that I am blessed to have been given the chance to see many friends with different personalities and characters in UCL. Especially, I would like to thank Mr B. J. Kim with whom I had memorable time during the first several years of my Ph.D. and Mr J. I. Kim and his family who never hesitated to help and comfort me whenever I had personal difficulties.

Finally, I would like to acknowledge the financial support on my tuition fees and living expenses by SaTReC, the Korea Science and Engineering Foundation(KOSEF), and the Overseas Research Scheme(ORS) Award.

Appendix A

Logic Implementation of the Sequencer and the Image RAM Addresser

This section presents the pin assignment of EPM7256E-12 to the sequencer logic and that of EPM5128 to the Image RAM Addresser logic. Their schematic diagrams at the top design level are also given. Figure [A.1] is the pin layout of EPM7256E-12 viewed from the bottom and Table [A.1] is its pin assignment to the sequencer. Figure [A.2] is the pin layout of EPM5128 viewed from the top and Table [A.2] is its pin assignment to the Image RAM Addresser. Figure [A.3] is the schematic diagram of the sequencer logic circuit at its top design level, Figure [A.4] is the schematic diagram of the BM_CNT function block, and Figure [A.5] is the schematic diagram of the Image RAM Addresser logic circuit at its top design level.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
U	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
T	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
R	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
P	o	o	o	o			o	o	o	o	o			o	o	o	o
N	o	o	o												o	o	o
M	o	o	o												o	o	o
L	o	o	o	o										o	o	o	o
K	o	o	o	o										o	o	o	o
J	o	o	o	o										o	o	o	o
H	o	o	o	o										o	o	o	o
G	o	o	o	o										o	o	o	o
F	o	o	o												o	o	o
E	o	o	o												o	o	o
D	o	o	o	o			o	o	o	o	o			o	o	o	o
C	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
B	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
A	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o

Figure [A.1] The bottom view of EPM7256E-12

PIN	NODE	PIN	NODE	PIN	NODE
T9	D0	H14	VCC	L17	M_ADD6
R15	D1	R14	VCC	U12	M_ADD7
U9	D2	D11	VCC	R13	M_ADD8
R9	D3	D14	VCC	B12	M_ADD9
J14	D4	P8	GND	A4	M_END
U15	D5	P10	GND	A8	M_RST
R10	D6	C7	GND	J2	M10
U14	D7	P15	GND	M1	M11
U13	D8	C13	GND	R2	OE_RAM
R12	D9	L4	GND	R3	PURGE
T13	D10	G14	GND	A5	RAM_BUFF_G
U10	D11	R4	GND	B9	RD_END
T11	D12	D4	GND	F3	REAL_END
T12	D13	H4	GND	K1	RE_RAM
R16	D14	R11	GND	C8	RST
U17	D15	D8	GND	U2	SHUTTER
T10	INTG_END	D10	GND	J16	WE_RAM1
U11	MASTER_CLK	D9	DAC_ADD0	J4	WE_RAM2
P9	STROBE	A7	DAC_ADD1	K2	WE_RAM3
P3	VCC	B8	DAC_ADD2	E2	WR_DAC0
P7	VCC	A9	DAC_BUFF_G	U6	WR_DAC1
C5	VCC	F1	DAC-ON/OFF	T5	WR_DAC2
K4	VCC	B1	HT_END	U7	WR_DAC3
P11	VCC	L3	M_ADD0	R7	WR_DAC4
C11	VCC	L1	M_ADD1	R6	WR_DAC5
G4	VCC	T6	M_ADD2	T7	WR_DAC6
L14	VCC	C9	M_ADD3	L2	WR_END
R5	VCC	G15	M_ADD4		
D7	VCC	K17	M_ADD5		

Table [A.1] The pin assignment of EPM7256E-12 to the sequencer

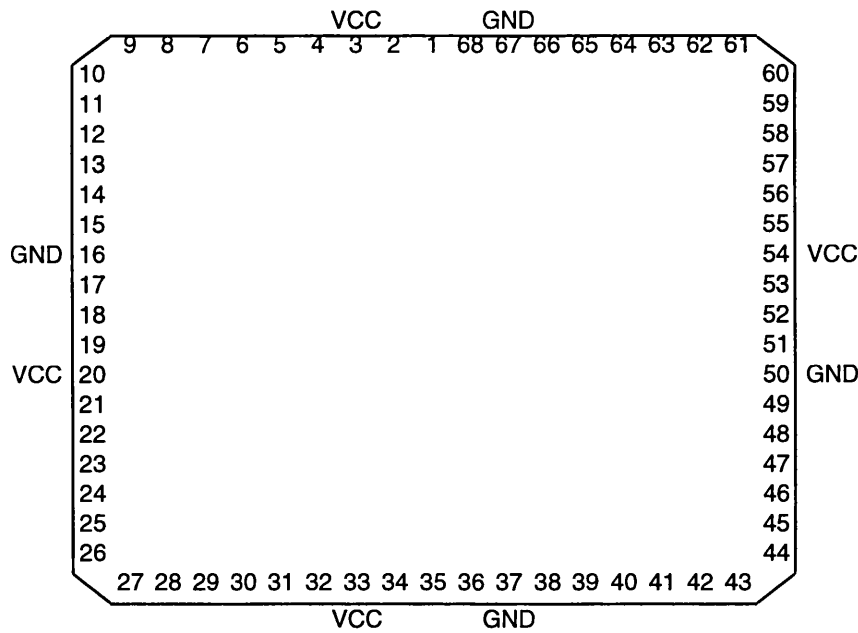


Figure [A.2] The top view of EPM5128

PIN	NODE	PIN	NODE	PIN	NODE
11	PA0	2	WE_CLK	43	ADD13
46	PA1	1	WR_STBE	44	ADD14
68	PA2	58	ADD0	45	ADD15
66	PA3	59	ADD1	24	MUX_A
36	PA4	60	ADD2	25	MUX_B
35	PA5	61	ADD3	26	RE_RAM1
34	PB0	62	ADD4	27	RE_RAM2
4	PB1	63	ADD5	28	RE_RAM3
5	PB2	64	ADD6	29	RE_RAM4
6	PB3	65	ADD7	30	WE_DAC1
7	PB4	38	ADD8	31	WE_DAC2
8	PB5	39	ADD9	52	WE_DAC3
9	PB6	40	ADD10	53	WE_DAC4
10	PB7	41	ADD11	55	WE_RAM
32	RD_STBE	42	ADD12		

Table [A.2] The pin assignment of EPM5128 to the Image RAM Addresser

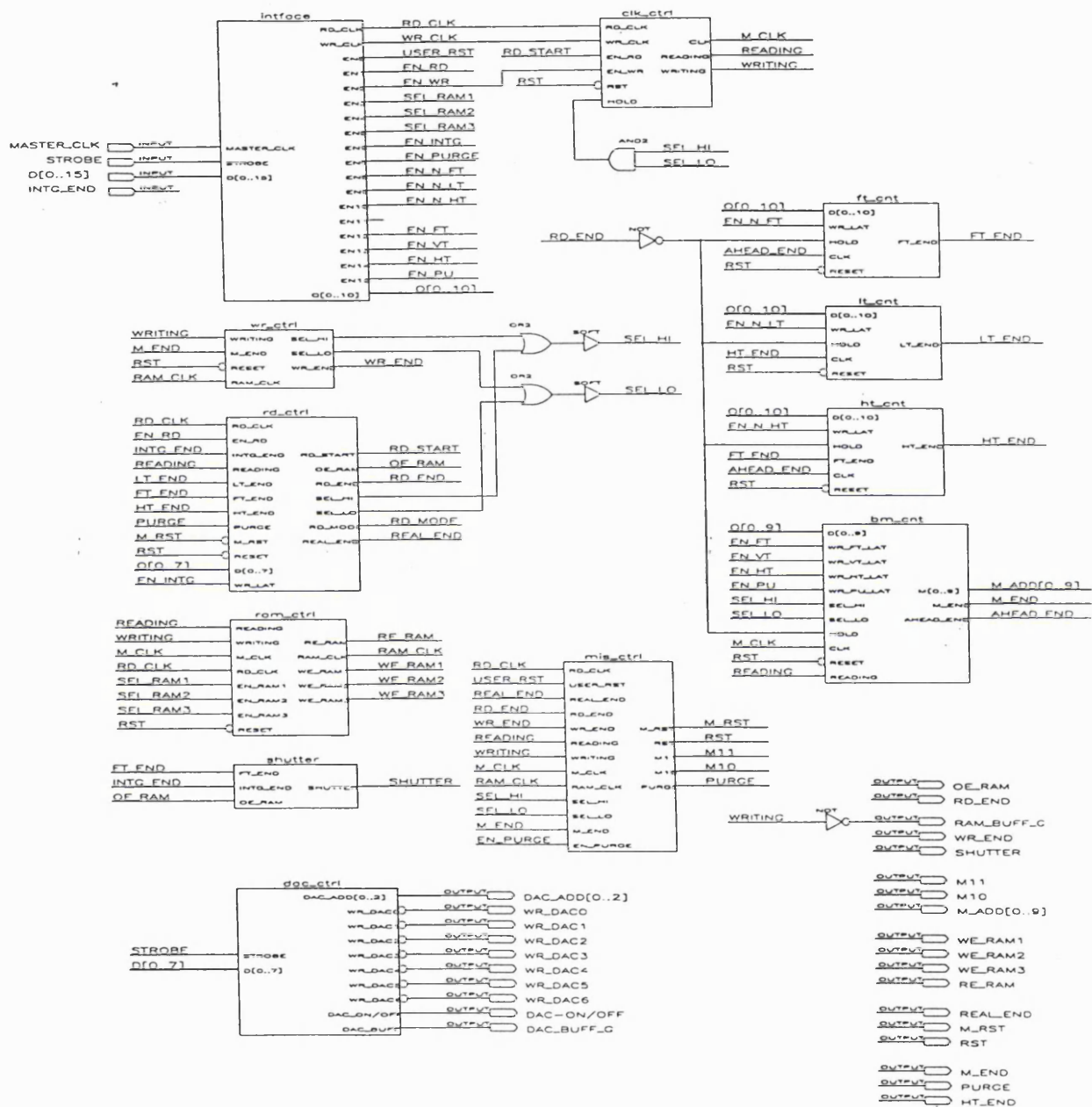


Figure [A.3] Logic circuit of the sequencer at the top design level



Appendix B

Class Header Files for the GUI and Control Software

This section presents the C++ header files where some classes for the GUI and control software are declared or some of them are defined.

B.1 GRAPHICS.HPP header file

```
#ifndef GRAPHICSHPP
#define GRAPHICSHPP

// common library header files
#include <iostream.h>
#include <fstream.h>
#include <sstream.h>
#include <conio.h>
#include <stdlib.h>
#include <cstring.h>
// super vga graphics header file
#include "svgacc.h"

typedef unsigned char byte;

// int type COLOR
typedef int COLOR;

// enumeration type BOOL
typedef enum {
    FALSE = 0,
    TRUE
} BOOL;

// enumeration type FLAG
typedef enum {
    OFF = 0,
    ON
} FLAG;

// struct BOX with 4 variables for coordinates
typedef struct {
    int x1;
    int y1;
    int x2;
    int y2;
} BOX;

// class Graphics handles graphics functions
class Graphics {
```

```

public:
Graphics();
virtual ~Graphics();

// PixelMode is declared in svgacc.h
// default value is SET (overwriting)
virtual void DrawDot(int, int, int, PixelMode = SET);
// 2 overloaded functions to draw box
virtual void DrawBox(int, BOX, PixelMode = SET);
virtual void DrawBox(int, int, int, int, int, PixelMode = SET);
// function to draw circle
virtual void DrawCircle(int, int, int, int, PixelMode = SET);
// 2 overloaded functions to draw line
virtual void DrawLine(int, BOX, PixelMode = SET);
virtual void DrawLine(int, int, int, int, int, PixelMode = SET);
// function to fill box
virtual void FillBox(int, BOX, PixelMode = SET);
// function to fill circle
virtual void FillCircle(int, int, int, int, PixelMode = SET);
// function to print string
virtual void PutString(int, int, int, int, string, PixelMode = SET);
// return font width & height
virtual int FWidth() {return FontWidth;}
virtual int FHeight() {return FontHeight;}
// return screen width & height
virtual int SWidth() {return ScreenWidth;}
virtual int SHeight() {return ScreenHeight;}

// RasterBlock is declared in svgacc.h
virtual byte SaveImage(int, int, int, int, RasterBlock *);
virtual void RestoreImage(int, int, RasterBlock *, PixelMode = SET);
virtual void Terminate(string = "");

protected:
int ScreenWidth, ScreenHeight;
int FontWidth, FontHeight;
int OldVideoMode;

virtual void RestoreMode();
};

// function to handle memory allocation error
void MemoryError(string);

// declaration of external variable to handle graphics
extern Graphics *GMaster;

#endif

```

B.2 EVENTS.HPP header file

```

#ifndef EVENTSHPP
#define EVENTSHPP

#include "graphics.hpp"
#include "mouse.hpp"

// define some of common key values
#define ESC 0x001b
#define ENTER 0x000d
#define BS 0x0008
#define TAB 0x0009
#define INSERT 0x5200
#define DELETE 0x5300
#define HOME 0x4700
#define PGUP 0x4900

```

```
#define END            0x4f00
#define PGDN           0x5100
#define LEFT           0x4b00
#define RIGHT          0x4d00
#define UP             0x4800
#define DOWN           0x5000
```

```
#define F1             0x3b00
#define F2             0x3c00
#define F3             0x3d00
#define F4             0x3e00
#define F5             0x3f00
#define F6             0x4000
#define F7             0x4100
#define F8             0x4200
#define F9             0x4300
#define F10            0x4400
```

```
// define EMode enumeration to combine mouse and keyboard inputs
```

```
// EMode == event mode
```

```
typedef enum {
    NO_ACT = 0,
    KEY,           // keyboard
    MOV,           // mouse movement
    LBP,           // mouse left button pressed
    LBR,           // mouse left button released
    LBD,           // mouse dragged with left button pressed
    RBP           // mouse right button pressed
} EMode;
```

```
// define struct EVE to hold EMode and
```

```
// values of key stroke and status of mouse
```

```
typedef struct {
    EMode mode;
    int key;
    int x;         // x coordiante of mouse
    int y;         // y coordinate of mouse
    int old_x;     // old x cood of mouse
    int old_y;     // old y cood of mouse
} EVE;
```

```
// class Events handles keyboard and mouse inputs
```

```
class Events {
public:
    Events();
    virtual ~Events() {}
```

```
    // ProcessEvents distributes events to functions
```

```
    // and returns struct EVE
```

```
    virtual EVE ProcessEvents();
```

```
    // mouse action functions
```

```
    virtual void ShowMouseCursor() { EMouse.Show(); }
```

```
    virtual void HideMouseCursor() { EMouse.Hide(); }
```

```
    virtual void EnableMouse() { EMouse.Enable(); }
```

```
    virtual void DisableMouse() { EMouse.Disable(); }
```

```
    virtual void SetMouseRange(int, int, int, int);
```

```
    virtual void SetMousePosition(int, int);
```

```
    // keyboard action functions
```

```
    virtual int GetKeyPressed();
```

```
    virtual int AnyKeysPressed();
```

```
    virtual void Delay(float);
```

```
protected:
```

```
    Mouse EMouse;
```

```
    int OldX, OldY;
```

```
    int Status;
```

```
};

// external variable declaration to handle events
extern Events *EMaster;

#endif
```

B.3 OBJECT.HPP header file

```
#ifndef OBJECTHPP
#define OBJECTHPP

#include "graphics.hpp"
#include "events.hpp"

// define enumeration ObjectStatus
typedef struct {
    FLAG Active;
} ObjectStatus;

// base class Object
class Object {
public:
    Object(int, int, int, int);
    virtual ~Object() {}

    // graphics related functions
    virtual void Display() = 0;
    virtual void Update(BOX) = 0;
    // status functions
    virtual BOX MyCood();
    virtual BOOL IsCoodIn(int, int);
    virtual ObjectStatus MyStatus() { return Status; }
    virtual void Activate() { Status.Active = ON; }
    virtual void Inactivate() { Status.Active = OFF; }

protected:
    int X1, Y1;
    int X2, Y2;
    ObjectStatus Status;
};

// base class ActiveObject
class ActiveObject : public Object {
public:
    ActiveObject(int, int, int, int);
    virtual ~ActiveObject() {}

    virtual BOOL Run() { return TRUE; }
    virtual void Exit() {}
    virtual void ProcessEvents(EVE) {}

    // graphics related function
    virtual BOX ClearArea();

protected:
    FLAG End;

    // events interface functions
    virtual void MouseMoving(int, int) {}
    virtual void LeftButtPressed(int, int) {}
    virtual void LeftButtReleased(int, int) {}
    virtual void LeftButtDragged(int, int, int, int) {}
    virtual void KeyIn(int) {}
};
```

```
// base class PassiveObject
class PassiveObject : public Object {
public:
    PassiveObject(int, int, int, int);
    virtual ~PassiveObject() {}

    // function to setup new coordinate
    virtual void NewCoord(int, int) {}

    // events interface functions
    virtual void LeftButtPressed() {}
    virtual void LeftButtUnpress() {}
    virtual void LeftButtReleased() {}
    virtual void LeftButtDragged(int, int) {}
    virtual void KeyIn(int) {}
};

#endif
```

B.4 ICON.HPP header file

```
#ifndef ICONHPP
#define ICONHPP

#include "object.hpp"

// base class Icon
class Icon : public PassiveObject {
public:
    Icon(int, int, int, int);
    virtual ~Icon() {}

    virtual void Display() {}
    virtual void Update(BOX) {}

    // function to setup new coordinate
    virtual void NewCoord(int, int);
};

// class IconList to handle list of Icons
class IconList {
public:
    IconList();
    ~IconList();

    // Icon linked-list operations
    BOOL Empty();
    int Size();

    void InsertItem(Icon *, int);
    void DeleteItem(int);
    Icon *Retrieve(int);

    Icon *First();
    void InsertFirst(Icon *);
    void DeleteFirst();

    Icon *Last();
    void InsertLast(Icon *);
    void DeleteLast();

    void AddItem(Icon*);

protected:
    // protected class ListItem
    class ListItem {
```

```
    public:
        Icon *ListItemValue;
        ListItem *ListItemLink;
        ListItem(Icon *icon, ListItem *link)
        {
            ListItemValue = icon;
            ListItemLink = link;
        };
        ~ListItem() {
            delete ListItemValue;
        };
};
// data member
ListItem *ListHead;
};

#endif
```

B.5 INPUTBOX.HPP header file

```
#ifndef INPUTBOXHPP
#define INPUTBOXHPP

#include "icon.hpp"
#include "text.hpp"

class InputBox : public Icon {
public:
    InputBox(int, int, int, string, string * = NULL);
    virtual ~InputBox();

    virtual void Display();
    virtual void Update(BOX);
    virtual void NewCood(int, int);

    virtual void LeftButtPressed();
    virtual void KeyIn(int);

protected:
    static byte INPUT_BOX_HEIGHT;

    COLOR border_clr;
    COLOR text_clr;
    COLOR background_clr;
    COLOR highlight_clr;
    COLOR unhighlight_clr;

    BOX TextBox;
    BOX ValueBox;
    string TextStr;
    string *ValueStr;
    string OldValueStr;
};

class IInputBox : public InputBox {
public:
    IInputBox(int, int, int, string, int &, int, int);
    ~IInputBox();

    void KeyIn(int);

protected:
    int &Value, Max, Min;

    string IToString(int);
    int STolInteger(string);
};
```

```

};

class FInputBox : public InputBox {
public:
    FInputBox(int, int, int, string, float &, float, float);
    ~FInputBox();

    void KeyIn(int);

protected:
    float &Value, Max, Min;

    string FToString(float);
    float SToFloat(string);
};

#endif

```

B.6 POPUPWIN.HPP header file

```

#ifndef POPUPWINHPP
#define POPUPWINHPP

#include "object.hpp"
#include "icon.hpp"
#include "text.hpp"

// class PopupWin derived from ActiveObject
class PopupWin : public ActiveObject {
public:
    PopupWin(int, int, int, int, string);
    virtual ~PopupWin();

    //own Rin & Exit function
    virtual BOOL Run();
    virtual void Exit();

    // any inherited class MUST provide these functions
    virtual void Display();
    virtual void Update(BOX);
    virtual BOX ClearArea();

    // functions to handle Icons
    virtual void AddIcon(Icon *);
    virtual void RemoveIcon(Icon *);
    virtual Icon *CurrentIcon(int, int);
    virtual byte NoIcon();

protected:
    static byte WIN_TITLE_HEIGHT;
    string Title;
    BOX TitleBox;
    BOX OldBoundary;
    // dragged distance when window is moved
    int win_dx, win_dy;
    // Moving flag
    FLAG Moving;

    // PIconList to store Icons for windows
    IconList *PIconList;
    // ActiveIcon is pointer to Icon on which mouse
    // left button is pressed
    Icon *ActiveIcon;

    // colors
    COLOR background_clr;

```



```
COLOR border_clr;
COLOR inactive_clr;
COLOR active_clr;
COLOR inactive_title_clr;
COLOR active_title_clr;

// user interface members
virtual void ProcessEvents(EVE);
virtual void LeftButPressed(int, int);
virtual void LeftButReleased(int, int);
virtual void LeftButDragged(int, int, int, int);
virtual void KeyIn(int);

// declare PWinExitButton class
// this Icon is only used by active window objects
// Exit Button at top left corner
class PWinExitBut : public Icon {
public:
    PWinExitBut(COLOR, COLOR,
                int, int, int, int, PopupWin *);
    ~PWinExitBut();

    void Display();
    void Update(BOX);
    void LeftButPressed();
    void LeftButReleased();
    void LeftButUnpress();

private:
    COLOR inactive_clr;
    COLOR active_clr;
    COLOR fill_clr;

    PopupWin *ParentWin;
};
// end of declaration of PopupWinExitButton

PWinExitBut *ExitButton;
};

#endif
```

B.7 MENU.HPP header file

```
#ifndef MENUHPP
#define MENUHPP

#include "object.hpp"
#include "text.hpp"

// simple class MenuFrame, one for each sub-menu
class MenuFrame {
public:
    MenuFrame();
    ~MenuFrame();
    void SetFrame(int, int, byte, string *);
    void Display();
    void Undisplay();
    void HighlightItem(byte);
    void UnhighlightItem(byte);

    byte Noltem;
    BOX FrameBox;

protected:
    COLOR fill_clr;
```

```

    COLOR border_clr;
    COLOR text_clr;
    COLOR highlight_clr;

    string *FrameStr;
    RasterBlock *SavedImage;
};

// declaration of MenuBar derived from Icon class
class MenuBar : public ActiveObject {
public:
    MenuBar(int, int, int, int, byte, string *);
    ~MenuBar();
    void MakeMenu(byte, byte, string *, string * = NULL);

    void Display();
    void Update(BOX);

    BOOL Run(int &, int &);
    BOOL Run() { return TRUE; }
    void Exit();
    void ProcessEvents(EVE);

protected:

    COLOR fill_clr;
    COLOR highlight_clr;
    COLOR text_clr;
    COLOR border_clr;
    int ActiveFrame;
    int ActiveItem;
    FLAG FrameOn;
    byte NoFrame;
    string *MenuStr;
    MenuFrame *Frame;

    void LeftButtPressed(int,int);
    void LeftButtReleased(int, int);
    void KeyIn(int);

    void Highlight(byte);
    void Unhighlight(byte);
};

#endif

```

B.8 BMEWIN.HPP header file

```

#ifndef BMEWINHPP
#define BMEWINHPP

#include "filewin.hpp"
#include "menu.hpp"
#include "commbox.hpp"

extern string Main[];
extern string File[];
extern string FileSK[];
extern string Edit[];
extern string EditSK[];
extern string Options[];

class BMEWin : public PopupWin {
public:
    BMEWin(string);
    ~BMEWin();

```

```
BOOL Run();
void Exit();
void Display();
void Update(BOX);
BOX ClearArea();

// MenuBar-related functions
void SetMenu(byte, string *);
void MakeMenu(byte, byte, string *, string * = NULL);

protected:
static byte MAX_NODE_NAME;
static byte MAX_NO_WAVE;
static unsigned MAX_NO_UNIT;
static byte BOX_HITE;
static byte MAX_MAG;
static byte DEFAULT_MAG;
static byte MIN_MAG;

COLOR box_clr;
BOX FilenameBox;
BOX InfoBox;
BOX NodeBox;
BOX WaveBox;

void MouseMoving(int,int);
void LeftButtPressed(int,int);
void LeftButtDragged(int,int,int,int);
void LeftButtReleased(int,int);
void KeyIn(int);

// special functions
void ReadInit();
BOOL FileRead();

// private display function
void DisplayNode();
void DisplayWave();
// overloaded function to display one waveform
void DisplayWave(int);
unsigned DisplayCoord(int);

// misc. action functions
void NewBitmapIndex(int);
void NewNodeIndex(int);
void Magnify(int);
void HighlightNode(int);
void UnHighlightNode(int);
void NodeBoxClick(int);
void ShadeWave(unsigned, unsigned);

// menu interface
void DoSomething(int, int);
void FileNew();
void FileOpen();
void FileSave();
void FileSaveAs();
void EditHighBM();
void EditLowBM();
void EditCopy();
void EditPaste();
void EditNode();
void OptionHome();
void OptionSize();
void OptionGoto();

// dialogue functions
void PutFilename(string, COLOR = BLACK);
```

```
void PutInfo(string, COLOR = BLACK);
void PutWarn(string, string = "");
string GetString(string, string, string = "");

// interface parameters
unsigned NoNode;           // # of nodes
unsigned NoUnit;           // # of resolution units
string *Node;              // array to store node names
unsigned long *Bitmap;     // first array to store Bitmap
unsigned BitmapIndex;
byte NodeIndex;
byte MagFactor;

fstream BMfile;
string Directory;
// *.bit Bitmap file name
string BMfilename;

FLAG Block;
unsigned BlockLeft, BlockRight;
FLAG Copy;
unsigned CopyLeft, CopyRight;
unsigned CopyNode;

MenuBar *Menu;
BOOL isThereMenu;
};

#endif
```

B.9 FILEWIN.HPP header file

```
#ifndef FILEWINHPP
#define FILEWINHPP

#include <dirent.h>
#include "popupwin.hpp"
#include "button.hpp"

// class PFileWin derived from PopupWin class
class PFileWin : public PopupWin {
public:
    PFileWin(int, int, string = "File Window",
             int = 100, int = 100);
    ~PFileWin();

    string Run(string, string);
    BOOL Run() { return TRUE; }
    void Exit();
    void DisplayDir();
    void Display();
    void Update(BOX);
    BOX ClearArea();

protected:
    static byte MAX_NO_FILE;

    DIR *Directory;
    string Dirname;
    string Filename;
    string Extension; // keep valid extension name
    string ReturnStr;
    string *DirStr;
    string *FileStr;

    COLOR text_clr;
```

```
    COLOR highlight_clr;
    byte NoDir, NoFile;
    byte OldRow, OldCol;
    byte BoxHeight;
    BOX DirnameBox;
    BOX FilenameBox;
    BOX DirBox;
    BOX FileBox;
    byte RowOffset;

    Button *UpButton;
    Button *DnButton;
    Button *OKButton;
    Button *CanButton;

    void ReadDir();
    // check if given name is directory name
    BOOL IsDirName(string);
    // check if file name is valid
    BOOL IsFileName(string);
    void DirSelected();
    // misc. functions
    void Highlight(byte, byte);
    void UnHighlight(byte, byte);
    void ShowDirName();
    void ShowFileName();

    void PageUp();
    void PageDn();

    void LeftButtPressed(int,int);
    void LeftButtReleased(int,int);
    void KeyIn(int);
};

void QSort(string *, byte);
int sortfnt(const void *, const void *);
#endif
```

B.10 SETUPWIN.HPP header file

```
#ifndef SETUPWINHPP
#define SETUPWINHPP

#include "popupwin.hpp"
#include "inputbox.hpp"

class SetupWin : public PopupWin {
public:
    SetupWin(string);
    ~SetupWin();

    virtual BOOL Run();
    virtual void Exit();

protected:
    Icon *OldIcon;

    virtual void LeftButtPressed(int, int);
    virtual void LeftButtReleased(int, int);
};

extern string BiasDAC[];
extern float BiasDACVal[];
extern float BiasDACMax, BiasDACMin;
extern float DACHMax, DACHMin;
```

```
extern float DACLMax, DACLMin;
extern string ClkDACa[];
extern float ClkDACaHVal[];
extern float ClkDACaLVal[];
extern string ClkDACb[];
extern float ClkDACbHVal[];
extern float ClkDACbLVal[];
extern string Para1[];
extern int Para1Val[];
extern int Para1Max[];
extern int Para1Min;
extern string Para2[];
extern int Para2Val[];
extern int Para2Max[];
extern int Para2Min[];
extern unsigned FTBU;
extern unsigned VTBU;
extern unsigned HTBU;
extern unsigned PUBU;
extern string Directory;
extern string FTFile;
extern string VTFile;
extern string HTFile;
extern string PUFile;

// application-dependent declaration of DAC SetupWin
class DACSetupWin : public SetupWin {
public:
    DACSetupWin();
    ~DACSetupWin() {}

    void Exit();
    void Display();

protected:
    BOX BiasBox;
    BOX DACaHBox, DACaLBox, DACbHBox, DACbLBox;
};

// application-dependent declaration of Parameter SetupWin
class ParaSetupWin : public SetupWin {
public:
    ParaSetupWin();
    ~ParaSetupWin() {}

    void Exit();
    void Display();

protected:
    BOX Box1, Box2;
};

#endif
```

Appendix C

Characterisation of a CCD Imaging System

The observed quantity in a CCD imaging system is the number of incident photons accumulated in each pixel whereas the measured quantity is a small output voltage amplified by an amplifier circuitry and digitised by an ADC. The gain of a CCD imaging system is generally defined as the ratio of the measured voltage in ADU to the number of electrons. If all physical parameters of an imaging system are known, it is straightforward to calculate its gain. However, it is usually obtained from experiments because neither all physical parameters are known nor they are constant over different operating conditions.

Any CCD imaging system is susceptible to various noise sources. In addition to photon shot noise, there are various noise sources intrinsic to CCD as described in sub-section 2.1.2 and other noise sources associated with the control system electronics. The amount of noise or the variance of the measured voltage gives the smallest number of incident photons that can be detected with an imaging system (i.e. noise-equivalent number of incident photons). Therefore, it is essential to evaluate an imaging system in terms of the amount of noise in order to extract some valuable scientific information from its measured data.

The overall gain of a CCD imaging system as a function of various physical parameters is presented in sub-section C.1. In sub-section C.2, a statistical testing method is introduced, which is widely used in calibration and analysis of the characteristics of a CCD imaging system in terms of gain and noise. The simple and practical mean-variance testing method that was used in the work of this thesis is described in sub-section C.3.

C.1 Gain of a CCD Imaging System

Ideally, the measured signal S_o in ADU from a CCD imaging system is a linear function of the number of electrons accumulated in a pixel given by

$$S_o = G N_e + b \quad [C.1]$$

where N_e is the number of electrons, G is the gain of the imaging system in ADU/e⁻, and b is a constant bias or offset in ADU. A constant bias is normally applied to the output signal in order to eliminate any conversion error that is possible when the signal is zero or slightly negative.

Therefore, the gain G must be known in order to obtain the number of electrons from the measured signal. It is a function of various physical parameters given by

$$G = \left(e^- / C \right) A_o \left(2^n / V_{fs} \right) \quad [C.2]$$

where C is the capacitance of CCD output floating diffusion, A_o is the overall amplification, V_{fs} is the full-scale input voltage of the ADC, n is the ADC conversion resolution. The first factor in equation [C.2] is called the sensitivity of a CCD and the last one is the conversion gain of an ADC. The overall amplification is given by

$$A_o = A_{CCD} A_{amp} \quad [C.3]$$

where A_{CCD} is the amplification by the CCD output MOSFET amplifier and A_{amp} is the amplification by the off-chip video signal processing electronics. Ideally, the number of incident photons N_p is related to the number of accumulated electrons by

$$N_e = \eta N_p \quad [C.4]$$

where η is the quantum efficiency.

C.2 Mean-Variance Testing Method

As pointed out earlier, the gain of a CCD imaging system is obtained experimentally because all physical parameters are hardly known. In addition, some of them are functions of various operating conditions. If any fixed-pattern noise such as pixel-to-pixel variation is removed from the measure signal, the overall noise from an imaging system n is given by

$$n^2 = p^2 + R^2 \quad [C.5]$$

where p is photon shot noise and R is the readout noise in rms electrons. If the squared gain is multiplied to equation [C.5], it gives

$$(G n)^2 = (G p)^2 + (G R)^2. \quad [C.6]$$

Because the left-hand side is the variance of the measured signal in ADU (V_m) and p in rms electrons is given by $\sqrt{S_m}/G$ where S_m is the mean of the measured signal, equation [C.6] becomes

$$V_m = G S_m + (G R)^2. \quad [C.7]$$

Therefore, if we calculate means and variances of the measured signal for different exposure levels and plot means along x axis and variances along y axis, the slope of the least-square fitted line gives the gain G and the square root of its y -intercept divided by G gives the readout noise R in rms electrons. This plot is normally referred to as the photon transfer plot and this method of deriving the gain and the readout noise is called the ‘mean-variance’ or ‘photon transfer’ method[75]. It must be noted that this method is valid only when the fixed-pattern noise and the bias are removed from the measured signal.

A typical mean-variance testing procedure is summarised as follows[75].

- 1) Obtain four or more flat-field image frames at a given exposure level.
- 2) Obtain a dark image frame at the same exposure level. A dark image frame is an image frame obtained with the CCD being blocked from illumination.
- 3) Select sub-areas of image frames that are free from artefacts and show the best uniformity.
- 4) Calculate the mean image sub-frame from all image sub-frames.
- 5) Subtract the mean image sub-frame from each image sub-frame.
- 6) Calculate the variance of each pixel from image sub-frames obtained in step 5) and calculate the mean of pixel variances over the selected sub-area (V_m).
- 7) Calculate the mean pixel value over the selected sub-area of the dark image frame. Subtract this mean value from the mean pixel value over the mean image sub-frame (S_m).
- 8) Repeat steps 1) - 7) for different exposure levels to draw the photon transfer plot.