DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

19980318 054

Aspex
Microsystems

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1995	Final Report

4. TITLE AND SUBTITLE

Near-Earth Object Detection Using the Associative String Processor

Final Rpt

5. FUNDING NUMBERS

F6170893W0939

6. AUTHOR(S)

Prof R.M. Lea

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Aspex Microsystems Ltd
Brunel University
Uxbridge, Middlesex UB8 3PH, UK

8. PERFORMING ORGANIZATION REPORT NUMBER

SPC-93-4059

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
PSC 802 BOX 14
FPO 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

SPC-93-4059

11. SUPPLEMENTARY NOTES

2 documents: Final report and requirements definition.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 words)

The NEO detection project at Aspex Microsystems Ltd was established to study the feasibility of providing a solution to the NEO problem through a pioneering Modular-MPC (Massively Parallel Computer).

14. SUBJECT TERMS

15. NUMBER OF PAGES

90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

European Office For Aerospace Research and Development (EOARD) : F6170893W0939

NEAR-EARTH OBJECT DETECTION

USING THE

ASSOCIATIVE STRING PROCESSOR

Final Report

Document Number: 323-002(neo02.wp5/net)

DTIC QUALITY INSPECTED 3

Issue: 1

Date of issue: 29 June 1995

Reproduction in whole or in part is permitted for any purpose of the United States Government

Aspex Microsystems Ltd
Brunel University, Uxbridge
United Kingdom, UB8 3PH
Tel : +44 (0)1895 274000 ext 2368, Telex 261173G
Facsimile : +44 (0)1895 258728

DISTRIBUTION LIST

Victoria Cox	(EOARD)
Alain Maury	(O.C.A)
Prof R M Lea	(Aspex)
J Lancaster	(Aspex)
Dr A Krikelis	(Aspex)
Rosh John Joseph	(Aspex)

NEAR-EARTH OBJECT DETECTION ON THE ASSOCIATIVE STRING PROCESSOR

The purpose of this document is to provide the final results of the study for the development of an ASP-based Near-Earth Object Detection System. This project is being jointly researched by Aspex Microsystems Ltd., Brunel University, England and the Observatoire de la Cote d'Azur, Nice, France.

Table of Contents

1

Introduction

Since 16th-22nd July 1994, when 21 fragments of the Comet Shoemaker-Levy 9 smashed into the jovian atmosphere at speeds of up to 60km/s, the destructive and explosive nature of the impact has stimulated growing interest and fears of a similar occurrence on Earth. Impact craters around the world and the moon has shown that the Earth too has not escaped from asteroid collisions. Impacts (e.g. the 1908 Tunguska incident) and near-misses (e.g. 1,100,000 km and 170,000 km of asteroids 1989FC and 1991BA respectively) of astronomical objects or near-Earth objects (NEOs) have, in recent years, manifested an urgency for the need of a collision awareness (and, possibly prevention) programme.

However, near-Earth asteroids are not the only cause of threat. For the past few decades, man has been sending more and more rockets and satellites into space for the purpose of discovery, communication, weather forecasting, military surveillance etc. As time goes by, these equipment would age into functional disuse while continuing to orbit the Earth and in some cases even gradually being pulled back by the Earth's gravitational field. This has resulted in a new threat, that of falling space junk. An NEO detection survey would also help safeguard against such eventualities.

As a first step towards achieving NEO detection and collision prevention goals, a comprehensive survey to compile a detailed inventory of the trajectories and physical characteristics of all asteroids in the near-Earth environments was necessary. Ofcourse, survey programmes of this type also have other spin-offs of geological and astrophysical significance. The composition of NEOs could give vital clues about the materials from which our Universe was formed. The difference in composition between various types of NEOs would also indicate how material had spread through the Solar System. The orbit of these objects would hint at how the debris escaped from the Solar System. In the future, NEOs could be a potential mining resource. A few near-Earth asteroids are easier to reach (in terms of their differential velocity) than the moon. Other areas that could benefit from the NEO detection program include artificial satellite tracking and space-based surveillance where algorithms and/or systems requirements are very similar.

NASA international workshops were organised in 1992, to discuss the possible threat of NEOs and suggest suitable future scientific research and possible strategic defence measures. It was recommended

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

that a concerted effort by the astronomical community through a globally co-ordinated long-term search programme with the prime objective of detecting and cataloguing these interplanetary fugitives. These recommendations have been stimulating interest in Europe. Indeed, the EUNEASO (EUropean Near Earth Asteroid Search Observatory) project recently established aims to provide a local network of French, German, Italian and Swedish observatories, as a first step towards a European contribution to a world wide future network, as proposed in the Spaceguard Survey.

With hind-sight and valuable empirical evidence from previous survey programs (PCAS, PACS, AANEAS, Spacewatch etc.), astronomers have realised that besides the need for larger CCD sensor arrays to be integrated into the imaging system, the unsuitability of current observatory facilities for NEO detection has been primarily a lack of sufficiently high performance processing workhorses, for the subsequent image processing. Computing performance rather than the optical system have been one of the main reasons for current detection systems in achieving a low discovery rate and are hence proving to be unsuitable in achieving a comprehensive survey of the skies.

Accordingly, the NEO detection project at Aspex Microsystems Ltd. was established to study the feasibility of providing a solution to the NEO problem through a pioneering Modular-MPC (Massively Parallel Computer). This study follows the application requirement study for NEO detection conducted by Aspex Microsystems Ltd. [JOS94a].

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

2
Application Analysis

The Spaceguard survey report [MOR92] has laid the foundation for most of the NEO detection research currently undertaken. The report recommended a network of 6 telescopes, each fitted with large format CCDs in a manner that utilises the focal plane efficiently, surveying at least 6000 sq.deg/month and detecting up to 30,000 objects/sq.deg.

The discovery rate is dependent on the number of available asteroids at a given limiting magnitude. and is proportional to the primary sky coverage. Primary sky coverage is governed by the telescope's field of view and the overall efficiency of the detector or imaging medium. Another consideration is the limiting magnitude of the telescope/detector assembly. A fainter threshold for detections seems desirable in that the search volume is increased and fainter objects are sampled. to achieve fainter thresholds, larger aperture telescopes and/or more sensitive detectors are required. A model of a whole sky survey for NEO detection [MOR92] revealed that to maintain discovery completeness and discovery rate without compromising warning times, large telescopes surveying large volumes of the sky at a limiting magnitude of less than 24 were necessary. And with even such compromises, survey periods of about 25 years were expected.

The detection of near-Earth objects involves the extraction of astronomical objects from images obtained from the CCD camera mounted on a telescope and the recognition of these objects to give a possible list of asteroids that could then be classified as Earth-approachers or Earth-crossers. Asteroids are basically detected by virtue of their constant velocity motion in space. Hence, these objects are best distinguished through comparisons of different frames of the same field, taken over a period of time. From these detections on different frames their orbital characteristics may be determined and hence decisions can be made on whether these detected objects could be possible near-Earth objects. Non-stationary objects and hence possible asteroids are seen as either streaks found in the images or by the apparent motion of objects between frames of the same field. The latter form of detection is the most likely criterion that would establish the presence of NEOs.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

2.1 The NEO detection system

Asteroid detection may be achieved with systems as shown in Fig. 2.1, comprising of 6 major components; namely the imaging unit, the acquisition unit, the system controller, the image archives, the detection engine and the host. This fundamental configuration is typical of most of the current survey telescope systems. Examples of such systems are already in evidence in the Spacewatch program at Kitt Peak, Arizona, USA and the NEO detection program at the Observatoire de la Cote d'Azur, Nice, France. Though the specifics might differ, their functionalities and principle are similar. Agreeably, the features of the sensor and the data acquisition are pivotal in the overall requirements for the system, hence justifying a larger allocation in the discussion, in this section[3].

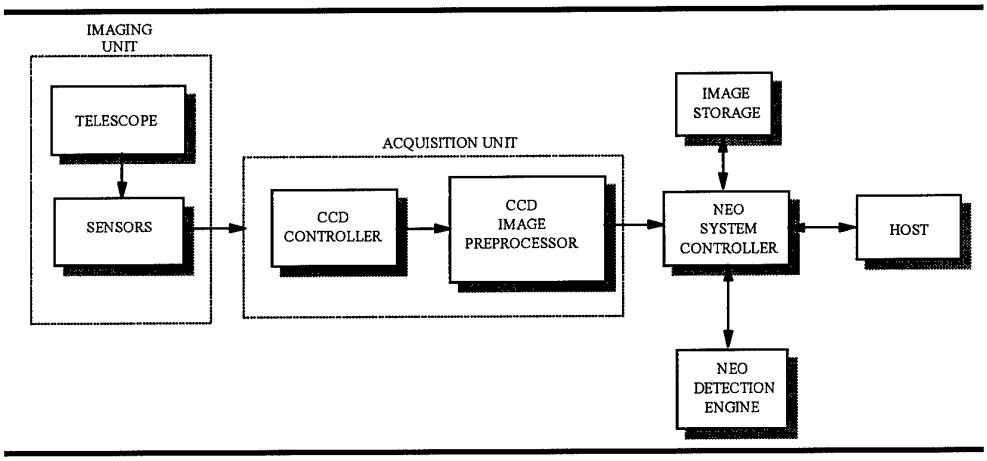

Fig 2.1 Functional Block diagram of the NEO detection system

A Schmidt telescope and photosensitive sensors make up the imaging unit. The wide aperture Schmidt telescope provide a large field of view as compared to traditional Newtonian telescopes, making it ideal for a survey. In the past, the photosensitive elements popularly used by astronomers were photographic elements. However, recently the use of photography has faced a descendency to charge coupled devices. A comparison of the detection characteristics of photography and CCDs [JAN87], in Table 2.1, reveal that this trend is not at all unjustifiable. CCDs have a higher detective quantum efficiency[4], higher operational detective quantum efficiency[5], better spectral and dynamic range, stability and linearity.

The CCDs are housed in the focal plane of the telescope. As ideal a detector as the CCD may appear to be, in the real world it does have some limitations. A major problem is that CCD sizes are limited to the largest wafers available and low manufacture yield for good quality large format CCDs. However, even

[3] Specific details mentioned about the system are mentioned with reference to the facilities provided for at OCA
[4] It provides a measurement of the amount of photons actually detected by a sensor, given the total amount of photons falling
[5]This gives the measurement of the detector efficiency with regard to actual image capture and the time after which useful information has been extracted and interpreted

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

the larger CCDs such as the 2k x 2k Loral, Kodak and Reticon CCDs, the 2k x 4k Loral CCD, the 4k x 4k Ford CCD, are too small to cover the focal plane.

Properties	Photography	Charged-coupled Devices
Resolution	High	Moderate
DQE	~10%	~80%
ODQE	Low	High
Spectral Range	Visual spectrum	X-ray to Near Infrared
Dynamic Range	~100	~100000
Stability	Poor	Good
Linearity	Poor	~0.1%
Photometric Accuracy	5%	0.5%
Surveying Efficiency	Moderate	Moderate

Table 2.1 Comparison of photography and charge coupled devices

Also, large format CCDs have prohibitively long read-out times and can make it cost-ineffective as it reduces overall telescope observation time. However, through the use of multiple channels for read-out, this limitation could be appeased. Read-out times can also be improved but at the expense of more noise. Techniques do exist to reduce these acquisition times while aiming to increase signal to noise ratios. Most CCDs now incorporate MPP[6] and Super-MPP modes [AST94] to reduce dark current noise. Also the use of dual-stage output amplifiers can increase read-out rates significantly e.g. for the EG & G Reticon CCD RA2000JAU-22X [WIN94] an optional two stage source follower increases data rate from a maximum of 2 Megpixels/sec to 15 Megapixels/sec. Another method is to exploit spectral characteristics of the scene to remove background intensity. Since a large portion of the light reflected by an asteroid is red, short wavelength cut-off filters could be used to filter out background intensity.

Astronomers have thus realised that to achieve effective utilisation of the imaging area in the focal plane, the solution is to create a large active CCD sensor area from a mosaic of smaller CCDs e.g. 30 CCDs in the photometric telescope used for the Sloane Digital Sky Survey [KEN92]. Previously CCD chips were fabricated and packaged such that there would exist some dead space bordering the active pixel array. Thus, only an N x M array with considerable dead space in-between or a chequered array [MAU93a], as shown in Fig. 2.2, were possible. But with the advent of edge-buttable devices, 1 x N and 2 x N[SEK92][LUP94][GEA91][STO94] configurations have become possible, with minimal dead space between the CCDs. Thus maximising the use of available imaging area in the focal plane.

Given the need to compare images of the same field, three schemes may be employed in the CCD arrangement using either a 2 x N chequered or 1 x N configurations. Ideally, 3 columns of CCD arrays

[6]MPP : Multi-Pinned Phase

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

should be used to be able to detect moving objects and ascertain whether that motion is of a constant velocity. However, such a column configuration would limit the detection to objects that move sufficiently fast to show differences in individual column images. Another method is to use two columns and for the third comparison a standard database is utilised of the astronomical objects in that area of the sky. A single column of CCDs, scanning over a period of time before repeating observations of the same field can also be used. Two or three such observations would be necessary for detection. The advantages of the three and two column configurations is that they allow a larger survey area to be covered per day. However, the need of database access during the detection process may prove to be an added overhead e.g. disk accesses, added functionality in either the software or hardware. Given the sensitivity of observations to weather and "seeing" conditions, the ability to cover a larger area per day appears an attractive option, though at present this may be a little expensive.

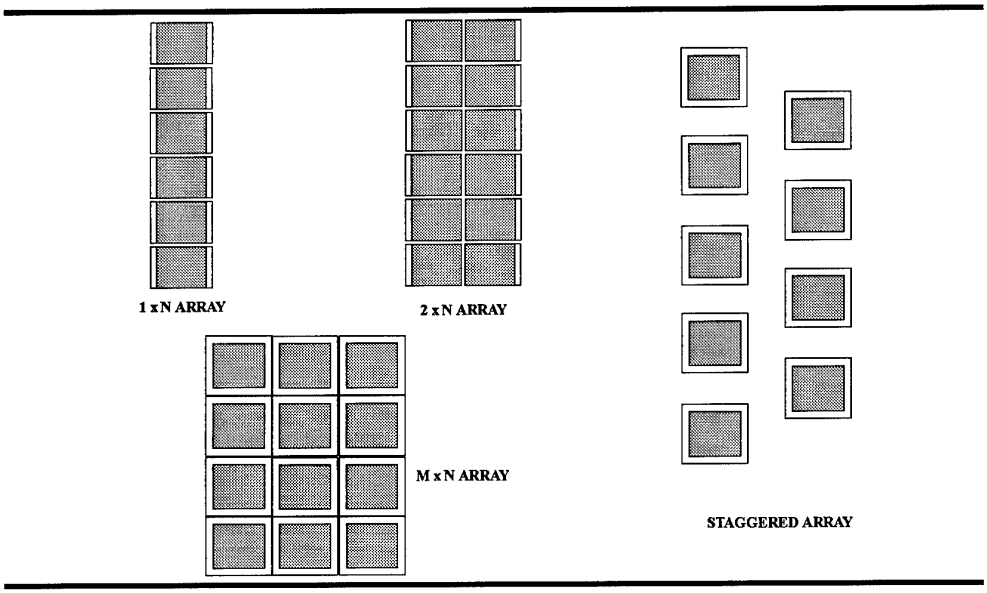

Fig 2.2. CCD configurations

The CCD can be operated in 2 basic modes; the stare and the sidereal scan. The stare mode is the simplest mode of operation for the CCD. Here the CCD is exposed to an area of the sky for a given period of time dependent on the type and limiting magnitude of the object observed. The CCD camera shutter is then shut and data is read-out while the telescope is repositioned for the next exposure. All the above configurations maybe utilised in this mode. The only difference is that individual CCD frame registration complexity would differ

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

In the sidereal (or drift scan or time-delay integration) mode, the rotation of the earth is utilised instead of repositioning the telescope. By clocking the CCD appropriately, charge packets collected in each pixel are transferred such that they move in unison with astronomical objects above them. This scan can be achieved without having to close the shutter, hence effectively making read-out time latent. However, there exist a directional aspect in this mode, read-out is only possible on the side of the CCD i.e. in the direction of scan. Thus only the chequered and the 1 x N configurations are useful. In this mode, the scan technique automatically removes any loss of image within the dead space for the chequered mode, unlike the 1 x N array. Also a larger observable area is possible.

However, the electronics of the acquisition unit functions much faster than the sidereal rate and hence are forced into wait states after a serial register[7] transfer. These wait states can be reduced by effectively increasing the sidereal rate. By moving the telescope in the direction of the sidereal direction, effective sidereal rate is increased. However, the price to pay is that read-out noise increases to levels that are sometimes unacceptable for low-level light scenes typical of astronomy. But this method appears to be the most suitable for a survey.

The acquisition process of data from the CCD to the detection engine is a complicated affair. The CCD requires multi-phase clocking to effectuate the transfer of charges. though with virtual phase clocking, in principle, a single phase clock would suffice. Timing and wave shaping circuits are necessary for these operation. These circuits are collectively termed as the CCD port as it provides the interface between the CCD and the rest of the system. The CCD port is also responsible for the analogue to digital conversion of the data. The CCD controller synchronises and governs the whole acquisition process. Several CCD controllers are now in existence ranging from discrete circuit assemblies [GUN87] at the Palomar Observatory, USA, DSP based [LEA88] at the Steward Observatory, USA and at O.C.A., France [MAU93b], transputer-based [WAL90] at the Greenwich Observatory, UK, programmable logic devices [HAN94] and even a combination of a transputer and a DSP [REI94] at the European Southern Observatory. However, their functionality is the same; providing the clocking sequences and voltage biases, but with varying programmability and flexibility. When a controller services more than one CCD, it should ensure the integrity of the data. It should also buffer the data before it is can be utilised by the detection engine. As in any acquisition system, a data back-up or archive facility is also provided. Due to the large data sizes, of the order of several Megapixels/sec, as shown in Table 2.1 comparing two CCDs,

[7] The serial register is the read-out register of the CCD array.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

associated with the survey it may be necessary to include another functional block for the compression of this data before storage.

With a large mosaic CCD imaging system, necessary for NEO detection, the acquisition control and synchronisation would overwhelm an single processor. Thus, suggesting a need for a hierarchical multi-controller schematic. Such a schematic would comprise of an individual CCD chip or a sub-array of CCD chips serviced by a local acquisition unit and controller, and a master controller then servicing a group of these units. This arrangement would imply the need for multi-channel controllers as seen in Fig. 2.3. With this scheme, the local controllers would be primarily responsible for controlling the acquisition of CCD data from the sub-array and also handle the temporary storage of individual frames in their individual local buffers. It would also handle the archiving of these images on secondary storage systems. The synchronisation of these local sub-array controllers and hence the acquisition of the entire array of the sensor would be managed by the array controller. The array controller would have to maintain the integrity of the overall composite image and allow access to the data by the detection engine.

	LORAL (2048x2048)				KODAK (2048x2048)			
Number Of CCDs Per Column	9				16			
Sidereal Speed	133 sec				79.8 sec			
Number Of Columns		1	2	3		1	2	3
Scan Speed (n x Sidereal Speed)	Acq Time/Pixel (microsec)	MegaPixels/sec			Acq Time/Pixel (microsec)	MegaPixels/sec		
1	31.7	0.28	0.57	0.85	19.0	0.84	1.68	2.52
2	15.9	.57	1.14	1.70	9.5	1.68	3.36	5.05
3	10.6	.85	1.70	2.55	6.3	2.52	5.05	7.57
4	7.9	1.14	2.27	3.41	4.8	3.36	6.73	10.10
5	6.3	1.42	2.84	4.26	3.8	4.20	8.41	12.61
6	5.3	1.70	3.41	5.11	3.2	5.05	10.0	15.1

Table 2.1 Data requirements for a telescope with a 30 cm focal plane

The detection engine now has the ominous task of reducing this massive amounts of data so that astronomers may be able to pin-point possible near-Earth asteroids. Given the large data volumes expected, the rapid simultaneous read-out from each CCD in the mosaic and the computation intensive algorithms typical of image processing applications, massive computing capabilities are required. The inherent parallelism in image processing make it suitable for massively parallel processing. In the application requirements [JOS94a] it was shown that given the massive data sizes, typical in astronomical imaging, even a simple convolution would require several Giga-OPS (OPS = 16 bit additions per sec).

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

With more advanced sensors available for data sensing, providing higher sensitivity and data rates of up to a few hundred megapixels per sensor, these requirements can indeed reach Tera-OPS.

Besides the requirements on processing performance, other important constraints also exists. As larger telescopes are built, larger and more advanced mosaics would be fitted in the focal plane. Processing requirements would increase by a few orders. Hence, the massively parallel processor should be scaleable to accommodate possible expansions of survey coverage and to meet the increase in processing performance requirements. Also as astronomers persevere to develop better and more complex algorithms to extract more precise information, the MPC should remain flexible and programmable. All these requirements suggest that the Modular- MPC approach can provide the necessary performance, flexibility and overall end-user acceptability.

A host facility in fig 2.3, gives the necessary interface to astronomers in the control of the entire NEO detection operation and hence, should be at the top end of such a system. Also through such a host, information regarding detected NEOs can be communicated to the astronomical community via existing global networks like the Internet. Such a facility would be essential when follow-up tasks are essential in confirming dubious objects by other observatories around the world.

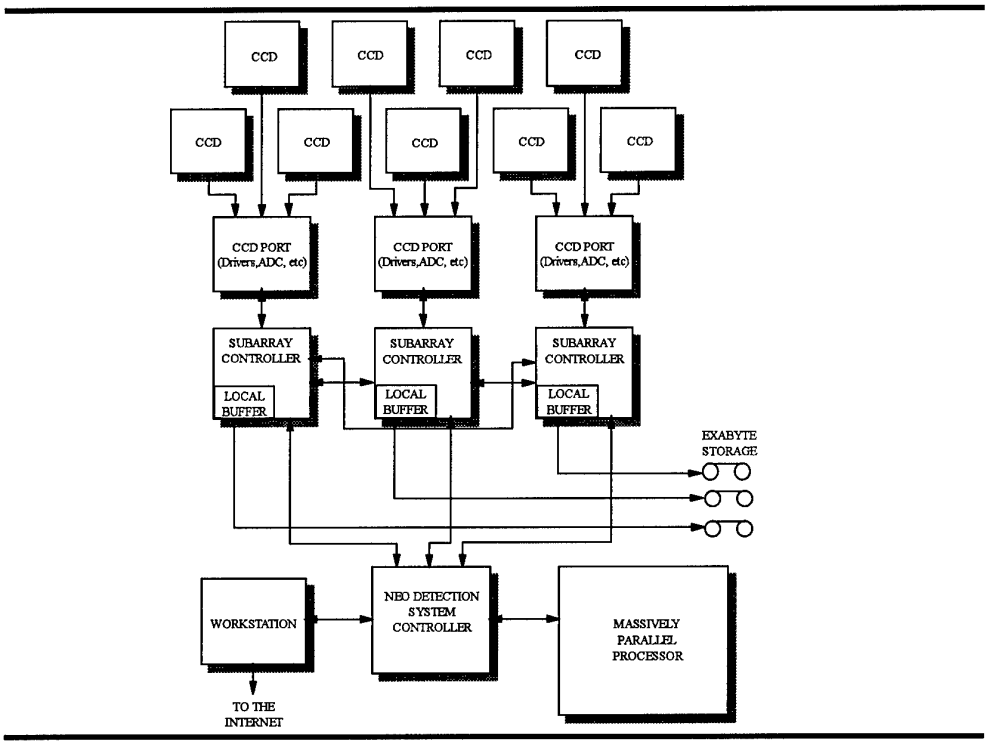

Fig 2.3 The NEO detection system

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

2.2 NEO detection

The NEO detection task sequence can be seen to be comprised of a task sequence as illustrated in Fig. 2.4. Image acquisition was referred to in the above section, and has been added in the task sequence to provide a more complete representation for the task sequence. The loop-back in the figure, elucidates the need for the repetition of image conditioning, object enhancement, segmentation, and analysis for repeated frames of the same field, during the scan sequence, in order to classify objects as NEOs. This is only the case when a single column of CCDs is used. For two or three column configurations such a loop-back would only be necessary when the field-of-interest for observation has changed. The Spacewatch program, distinguishes 2 stages only: streak detection and motion detection. Streak detection includes image conditioning to analysis tasks while object detection is achieved in the detection task.

2.2.1 Image Conditioning

Image conditioning can be seen as a task necessary to remove characteristics of the optical and sensing system that has biased the process of image capture. Algorithms include CCD reduction [VAL88][MAS89] and image restorative algorithms.

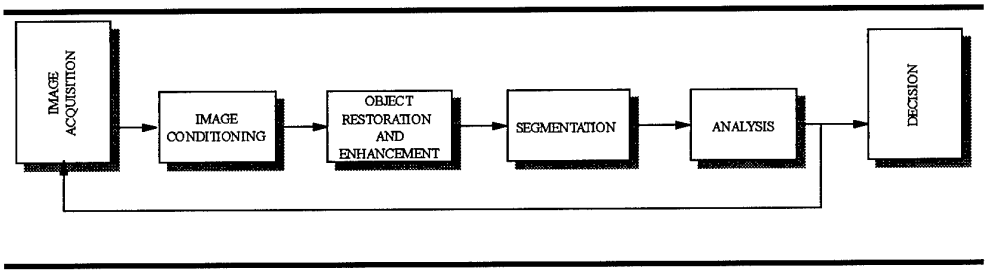

Fig 2.4. The NEO detection task sequence

CCD related noise are additive and/or multiplicative. Standard CCD reduction operations include :

i. **Row/ Column interpolation** : The replacement of bad columns and lines due to manufacturing defects in the CCD chips. The interpolation from neighbouring columns and rows is usually used to solve this.

ii. **Flat fielding** : This removes individual pixel-to-pixel sensitivities. Flat field exposures may be obtained by exposing CCDs to an illuminated white spot in the dome or for the purpose of good photometry this images taken at twilight are more appropriate. However, care should be taken as

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

illumination corrections may be necessary especially with the former. Once the flat field has been obtained, it can be used for all observations made on the day.

ii. **Dark fielding**: Some amount of dark current will always exist. This provides a correction for hole-electron pairs that are thermally generated. Dark fielding involves the subtraction of a bias level from the raw data. The dark fielding may be achieved by two methods; by utilising data from the overscan or prescan columns or lines, or by obtaining a dark image with the CCD shutter and/or telescope dome closed. Both methods provide certain compromises. The former method provides an approximation as it involves a calibration with data acquired simultaneously with the rest of the image. The latter is achieved using a dark field image acquired before the survey and is representative of localised dark current generation at the time the dark field image was taken.

Other operations include subtraction of a zero level using a zero length exposure calibration image to remove preflash-induced non-zero initial counts, division by an illumination image to correct illumination errors while obtaining the correcting field images.

2.2.2 Image Enhancement

Most image processing solutions are based on an approach whereby an image model is chosen to best represent the characteristics of the object of interest to aid in its detection or recognition. Most objects in an astronomical image are only a few pixels and are characterised by a gaussian-like intensity profile rising above the average background intensity. Consequently, astronomical models have been based on these characteristics.

The classical astronomical model is based on an image, $I(x,y)$, that is constituted by a slowly variable background with superimposed small scale astronomical objects which are associated to a point of maximum intensity. Then,

$$I(x,y) = ap(x,y) + \mu \qquad \text{(Eq. 4.1)}$$

where a is the amplitude of the object. $p(x,y)$ is the profile of the model object and μ is the mean of the background variation

Astronomical sources are defined by their radial profiles and are characterised by a circular, bivariate gaussian profile. Hence,

$$p(\Delta x, \Delta y) = e^{\frac{-(\Delta x^2+\Delta y^2)}{2\sigma^2}} \qquad \text{(Eq. 4.2)}$$

where $(\Delta x, \Delta y)$ is the distance from the centroid of the object, and σ is related to the size of the astronomical object taken as the FWHM ($\frac{FWHM}{2.36}$) over the range $-3\sigma < \Delta x, \Delta y < 3\sigma$

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

This is derived from the consideration that stars and other astronomical sources are point sources or have a dirac function. Most object detection projects, e.g. DAOPHOT[8], NEO Project at O.C.A, SDSS[9], have adopted this model.

Since asteroids and star-like objects are in the high frequency range, enhancement may be achieved by using high pass filters which enhances the resolution features and suppresses long wave fluctuations. However, the highest frequencies are also the most affected by noise. Therefore, these filters also enhance these noise components.

The filter for an astronomical image would then have to be a bandpass filter, p'(x,y), such that the restoration would provide the amplitude, a or a value as close to it as possible. Thus,

$$\Sigma\Sigma p'(x,y)I(x,y) = a \qquad \text{(Eq. 4.3)}$$

At O.C.A., the value of $p'(x,y)$ is determined by the method of least squares using a gaussian model for $p(x,y)$ to give a matched filter (Eq. 4.4) in the shape of a "Mexican Hat" (see Fig 2.5)[BIJ92], while DAOPHOT uses a truncated gaussian function such that its integral gives zero.

$$p'(x,y) = p(x,y)\left(\frac{S_0 p(x,y)) - S_1}{\Delta}\right) \qquad \text{(Eq. 4.4)}$$

where $S_0 = \Sigma\Sigma p(x,y)$, $S_1 = \Sigma\Sigma p^2(x,y)$, $S_2 = \Sigma\Sigma p^3(x,y)$ and $\Delta = S_2 S_0 - S_1^2$

Since the gaussian model (p(x,y)) is dependent on object size, taken to represent "seeing" conditions, these filters must be adapted to each exposure, based on an estimated full width at half magnitude (FWHM).

Detections due to non-astronomical objects such as cosmic ray events, should be removed at this stage also. The median filter may be used for this.

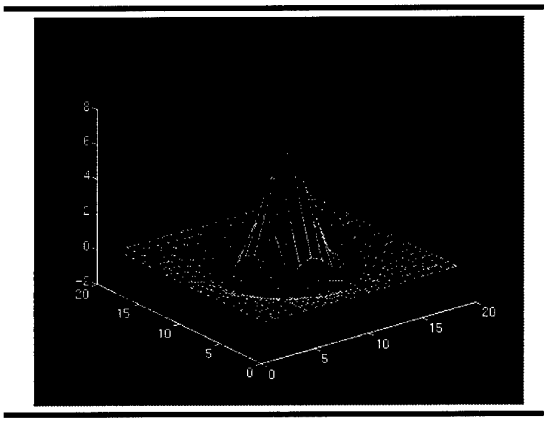

Fig 2.5 The "Mexican Hat" filter

[8]DAOPHOT : **D**ominion **A**strophysical **O**bservatory program for crowded field stellar **PHOT**ometry
[9]SDSS : Sloane Digital Sky Survey

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

2.2.3 Segmentation

The task of segmentation involves the separation of an input image into its constituent parts or objects such that objects of interest can be labelled and differentiated from other objects and from background pixels. This is very much dependent on the characteristics of the image, hence, the need for the previous steps to enhance these characteristics. Objects need to be separated as individual objects before they can be analysed to be able to make a decision. Ideally, pixels belonging to the same object have to be connected to form a single object definition.

Segmentation can be achieved sufficiently by thresholding. The threshold value can be determined by considering the fluctuations in amplitude due to the filtering operation. Since, the amplitude after filtering is given by Eq. 4.3. Then

$$\sigma^2(a) = \Sigma\Sigma\sigma^2(I(x,y))p'(x,y)$$

$$\sigma^2(a) = \mu\Sigma\Sigma p'(x,y) = \mu I$$

if a Poisson distribution of noise is considered. For effective detection, the threshold should then be set at approximately,

$$a > k\sigma(a)$$

with k3. Hence,

$$a > 3\sqrt{\mu I}$$

Values of k may be taken dependent on the required error margins. Higher values of k may result in detection failures (failure to detect faint objects that could be classed as NEOs) while lower values may result in detection errors (detection of false objects).

After filtering for stars or star-like objects, the enhanced central points are equivalent to the points of maxima on each object. All determined points of maxima in the full image can then be considered and labelled as a possible asteroid. These points of maxima should be determined as accurately as possible, to about 1/100th of a pixel, since even minute errors could introduce significant errors during astronomical co-ordinate reduction and on follow-up operations. Other characteristics are also used. DAOPHOT uses a sharpness and roundness criteria while FOCAS uses a count for the number of pixels within the star. These methods can be useful in reducing non-stellar detections including cosmic ray elements and many cosmetic flaws which tend to be either narrower than a seeing-broadened stellar image. Roundness may

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

also isolate extended objects due to overflow columns or rows from grossly over-exposed objects in the frame.

Fast moving asteroids may be detected as streaks thus, operators like the Hough transform could then be utilised for the analysis.

2.2.4 Analysis

At this point, processing now concentrates on objects rather than pixels. The task of analysis provides a proper representation and description of the detected objects for subsequent processing. Parameters of interest in the field of astronomy for the description of a objects are those related to object intensity magnitude, object intensity profile characteristics and object location.

Object intensity magnitudes can be represented in conjunction with its intensity profile by determining the second order geometric moments about the centre (given by the point of maximum at x_c,y_c) of the detected object. The geometric moments of order (p+q) of f(x,y) are defined as

$$M_{p.q} = \Sigma\Sigma(x-x_c)^p(y-y_c)^q I(x,y)$$

where x and y are the co-ordinates of the object's pixels in the image.

The object profile may be represented by determining the major and minor axes of the object's intensity profile. This may be determined by solving for the following equation,

$$\frac{A}{B} = \sqrt{\frac{\dot{x}^2}{\dot{y}^2}}$$

$$\pi AB = \sqrt{16\dot{x}^2\dot{y}^2}$$

where $\dot{x}$ and $\dot{y}$ are the moment centroids of order 2 in x and y

$$\dot{x}^2 = \Sigma_x \Sigma_y (x-\bar{x})^2 I(x,y)$$

$$\dot{y}^2 = \Sigma_x \Sigma_y (y-\bar{y})^2 I(x,y)$$

A simpler method to determine the centroid use an equation of the form ax^2+bx+c, to locate the points of maximum intensity and is given by a function of (a,b,c). Other intensity profile descriptors include ; other and higher-order moments, the aspect ratio : the ratio of the objects major axis length to the minor axis length, asymmetry : the rms fractional change in an object's intensity profile when reflected through its minor axis, uniformity : rms deviation of an object's intensity profile, along its major axis, from a template object of the same size and the same mean intensity, goodness to fit : the rms deviation of the object's intensity profile from a template object of the same size, orientation and peak intensity,

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

normalised to the peak intensity of the object. Most of these parameters have been used by the Spacewatch program, FOCAS, and by the NEO project at O.C.A..

Additionally, DAOPHOT and FOCAS uses synthetic aperture photometry also to remove background intensity before thresholding. Two annular regions are taken around a detected object. The first synthetic aperture, equivalent to the FWHM of the object, measures the total brightness dues to that stellar object. The other is taken several times larger than the first and measures the background intensity as the mode of that region. This background intensity after appropriate multiplication by the total number of pixels with in first region, is then subtracted from the total brightness to correct for all other sources other than the stellar object.

2.2.5 Decision

Decisions resulting from recognition of possible NEOs are achieved by the comparison of images from the same area of sky to extract objects that could be an asteroid. Asteroids can be differentiated by their motion at almost constant velocity through space. Hence, at this stage a further classification of detected objects can be performed by marking static and non-static objects which can then be further classified as asteroids or dubious objects.

Before classification can be undertaken, errors due to telescope pointing errors and atmospheric refraction will not cover precisely the same region of sky, and hence, would have to compensated. A stationary object will not appear at precisely the same location in an image due to these errors. However, the change in position is the same for each stationary object in the same scan. Registration can be achieved by determining the offset for a known stationary object in each frame. By generating a list of distances from the 1st and 2nd object lists from which a median can be obtained.

Classification may be achieved either by the use of a database of symbolic data referring to properties of detected objects or by an image comparison. The use of databases offer a considerable advantage in that it offers a reduction of processed data from numerical to symbolic data. However, with image comparison methods redundant information present in object-less regions result in unnecessary overheads in processing.

Whichever the scheme, the basic recognition operations are similar. The main objective of the algorithm is to determine motion in objects and then verify if this motion is achieved at constant velocity. This is achieved by a comparison of two to three separate frames of the same field, taken after given a period of time, to locate objects that are present in all three and are steadily moving across the field.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Firstly, lists of detected objects in frames are created. Objects that are not present at similar locations in either of the two lists of detected objects are classified as non-static objects. The search area to be considered is made on the premise that since moving asteroids are assumed to have a constant velocity, the change in co-ordinates should be more or less a constant and the search area should increase linearly from the second the third frame. The window within which this search is made is dependent on the maximum asteroid speed expected and the time lapse between exposures. On comparison with the list generated from a third image of the same area, differentiation of possible asteroids from other dubious objects can be made. Here the search is within a smaller window as the expected region is now known after the first comparison. When only two scans of the same field are obtained a standard star catalogue[10] may be used to isolate non-stationary objects. The second frame can then be used to determine whether the object was moving at constant velocity.

The orbital elements of these recognised asteroids are calculated and then used to further classify them into specific classes of asteroids e.g. Amor, Aten, or Apollo etc.

[10]Star catalogue : catalogued collection of all known stars

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

3

Modular-MPC architecture

An Associative String Processor(ASP) based modular massively parallel computer (Modular-MPC) is currently being studied as an architecture that is suitable to meet the requirements of NEO detection. The ASP architecture[LEA88] has proven its cost-effectiveness and performance-beating support of image processing applications [KRI91] while exploiting state-of-the-art microelectronics technology. ASP exploits the opportunities presented by the latest advances in the VLSI-to-ULSI-to-WSI technology trend. It also makes use of the continually improving high density system assembly techniques. ASP remains independent of technology, so it can benefit from the inevitable improvement in microelectronics technology without architectural modification.

3.1 Modular-MPC concept

Image processing applications are inherently data parallel and are well suited to massively parallel processing processing. However, despite the promising potential of current MPCs, architectural limitations have led to constraints in the integration of image processing applications, as MPCs have lacked the flexibility to match task program complexity and data evolution. This has led to the development of second generation architectures that utilise SIMD and/or MIMSIMD configurations for task pipelining and/or task parallel application solutions through multiple task modules.

The Modular-MPC strategy has been to develop an architecture that provides

i. application flexibility

Machine versatility as well as *performance scalability* achieved through application-specific configurations of generic hardware modules and software modules in order to match the natural parallelism of the application.

User acceptability gained by providing many familiar user-environments which adapt to the specific needs of different users.

ii. cost-effectiveness

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Size, weight, power and *cost* are, compared to current MPCs, significantly reduced through the consequent use of microelectronics. Application specific configurations are configured from *generic* modules, which can be mass-produced and are based on mass-produced microelectronics components (i.e. RAMs, microprocessors, FPGAs and ASP modules)

Operational efficiency is secured through Modular-MPC functionality and the efficiency of ASP modules

Future proofing of the Modular-MPC is "inherited" from the technology road-map for high-volume off-the-shelf components (e.g. memories, microprocessors, FPGAs) and ASP module technology upgrades.

3.2 Modular-MPC methodology

Most image processing applications require the execution of a sequence of *task packages*. *Tasks* control the evolution of specified sub-images and, typically, are application-specific in nature; the more complex usually comprising a hierarchy of *composition-tasks* and, at the lowest level *base-tasks*. *Base-tasks* are executed as a sequence of general-purpose *processes*, which control the navigation and evolution of specified data structures, as indicated below for typical *process* examples:*Processes* are executed as a sequence of primitive *operations* (e.g. +, -, x, and, or, <, =, > etc.) which control the evolution of pixels.

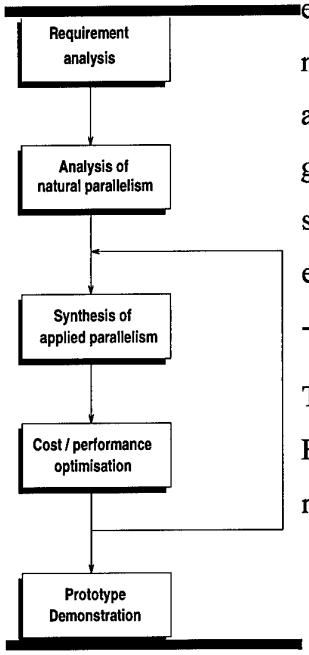

Fig 3.1 Modular-MPC Methodology

The Modular-MPC methodology is based on the following steps, shown in Figure 3.1, carried out for the particular requirements (e.g. frame size and frame rate) of an application.

Analysis of the natural parallelism *Task packages*, *tasks* and *processes* are identified. A flow-graph exposes opportunities for control-level parallelism. Subsequently *sub-images* associated with *task packages* and *tasks* as well as *data-structures* associated with *processes* are identified. The size of the *sub-images* and *data-structures* exposes opportunities for data-level parallelism.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Synthesis of the applied parallelism Modular-MPC software modules are installed in order to match the identified *tasks* and *processes*. The algorithm is then functionally verified on a general Modular-MPC. Subsequently a configuration of hardware modules for a specific Modular-MPC which matches the natural parallelism is derived and optimised.

3.3 Hardware architecture

Figure 3.2 shows the high-level architecture of the Modular-MPC. It is partitioned into three main functional blocks:

Massively Parallel Processor (MPP), where parallel processing takes place

Data Stream Manager (DSM), which supports parallel data transfer

Instruction Stream Manager (ISM), which supports sequential data transfer and control

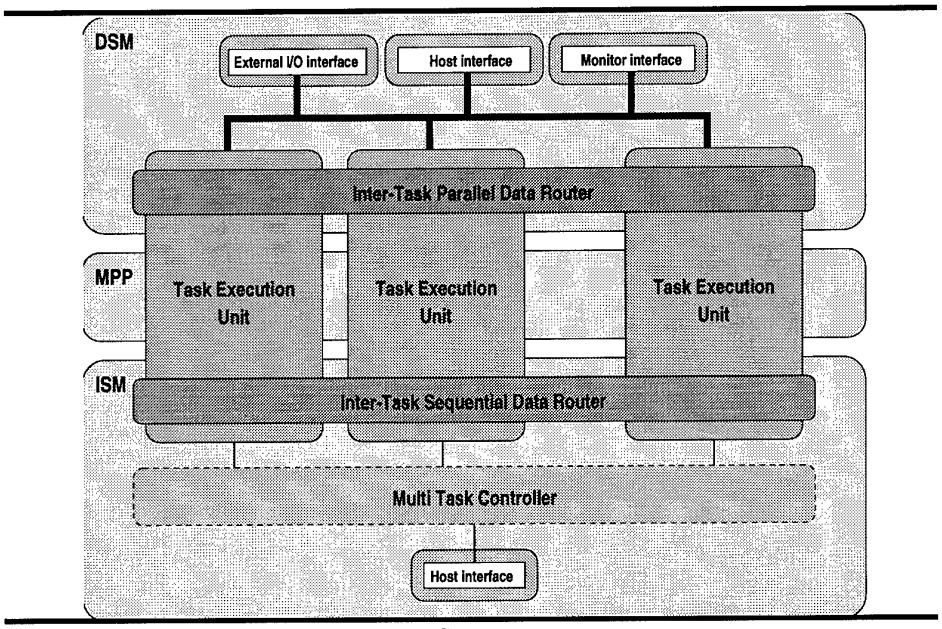

Fig 3.2 High level architecture of the Modular-MPC

It can be observed that the natural parallelism of image processing applications is often characterised by a massive data-level parallelism (e.g. all pixels in a *sub-image* can be processed in parallel) and a modest control-level parallelism between *task-packages* or *tasks*. The common MPC architectures, MIMD and SIMD, exploit either control-level or data-level parallelism, but cannot exploit both forms of parallelism simultaneously. make use of the respectively other level of parallelism. Therefore, Multiple Instruction

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

control of Multiple SIMD (MIMSIMD), which can exploit massive data-level parallelism as well as the modest control-level parallelism has been employed for the Modular-MPC.

A SIMD ensemble of PEs implement a Task Execution Unit. Hence, the MIMSIMD configuration is implemented by a number of Task Execution Units (TEU), each executing *task packages*. Depending on the control-level parallelism, the hierarchy of *tasks* in *task-packages* can either be executed sequentially in one Task Execution Unit or be spread over several TEUs. In the latter case, modest control-level parallelism is exploited by a number of TEUs working in parallel. However, the actual number of TEUs depends on a cost-effective compromise between temporal and spatial parallelism to achieve the minimum cost for given performance requirements. Finding this compromise requires balancing.

In order to exploit the control-level parallelism, a multi-task controller generates a private control stream for each Task Execution Unit (TEU). Task Execution Units are connected via two routers. Parallel Data can be exchange via the Inter-Task Parallel Data Router (ITPDR0, sequential data is exchanged via the Inter-Task Sequential Data Router (ITSDR).

Finally, TEUs can communicate with the outside via a number of interfaces. While sequential data can only be exchanged with the outside via the host interface which connects the Modular-MPC with the host workstation, a choice of interfaces is available for parallel data I/O:

host interface : In cases of moderate performance requirements for parallel data I/O, the transfer can be handled by the host workstation via the host-interface.

external I/O interface : For high-speed parallel data I/O and for specific data transfer protocols (e.g. HiPPI, SCSI) a number of external I/O interfaces is provided.

monitor interface : The specific requirements of parallel data output to high-resolution displays is handled by a monitor interface, which provides all functionality necessary (e.g. frame store, D/A converter) to directly interface to a monitor.

While the control-level parallelism is exploited by several Task Execution Units (TEUs) working in parallel, the data-level parallelism is exploited within each TEU, each of which implements a SIMD structure. The remainder of this section is devoted to the introduction of a TEU and its functional blocks.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

3.3.1 Task Execution Unit (TEU) overview

The TEU is based on ASP modules. Figure 3.3 shows a high-level view of the three main parts of a Task Execution Unit (TEU). The Massively Parallel Processor implements a Parallel Process Execution Unit (PPEU) based on the Associative String Processor (ASP). The ASP has been specifically designed for and has successfully demonstrated

machine versatility, which is inherent in its architecture

performance scalability due to the infinite scalability of the ASP string

reduction in *size*, *weight*, *power* and *cost* due to the fact that the string topology of the ASP architecture has been specifically developed for the use of microelectronics (VLSI, WSI) and packaging (MCM) technologies

operational efficiency, which is inherent in its architecture

future proofing, through regular advances in microelectronics.

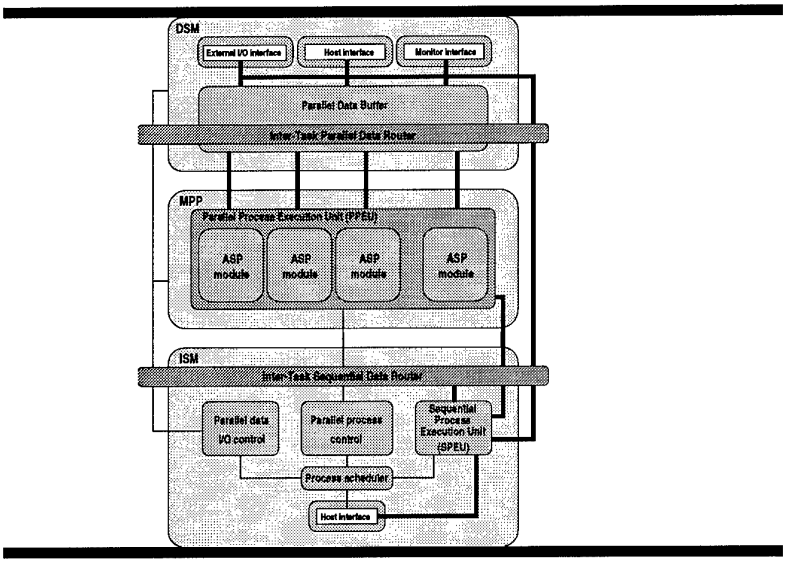

Fig.3.3 Modular-MPC Task Execution Unit

3.3.2 The Massively Parallel Processor

The Massively Parallel Processor (MPP) array is a string of identical associative processing elements (APEs) as indicated in Fig 3.3. Each APE (as shown in Fig. 3.4) incorporates an 64-bit data register and an 6-bit activity register, an 70-bit parallel comparator. Moreover, an APE includes a single-bit full-adder and four

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

status flags; the arithmetic carry(C), match and destination flags (M and D) and the activation flag(A). An APE also includes control logic for local processing and communication with other APEs. The APE can operate in three different data modes. The 64 bit data register can be configured for

storage and bit parallel processing of two 32-bit binary words

storage and bit-parallel processing of four 8-bit ternary byte fields

storage and bit-serial processing of one to three ternary contiguous bit fields of varying length (no more than 64 bits per field)

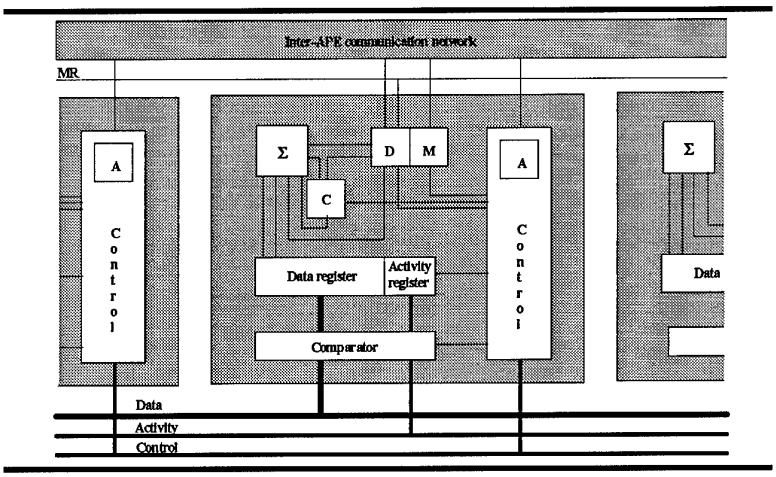

Fig. 3.4 The Associative Processing Element (APE)

Each APE connects to an inter-APE communication network(IACN), which runs in parallel with the APE substring implemented as a shift register and a chordal ring. It can be dynamically reconfigured, thus providing a cost-effective emulation of common network topologies. As an activity-passing, rather than a data-passing, network it minimises data transfers. The chordal ring enables the IACN to be implemented as a hierarchy of ASP substrings. Thus, communication times are significantly reduced through automatic bypassing of those ASP substrings which do not include destination APEs. In a similar way, namely through bypassing of faulty ASP substrings, fault-tolerance of the ASP modules is guaranteed.

All APEs share a common bit-parallel data (called the scalar data bus (SDB)), activity and control buses, and one feedback line called the Match Reply(MR). Through the Sequential Data and Control Interface, an external controller (the Instrunction Stream Manager) maintains the buses, feedback line, and Link Left and

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Link Right ports of the inter-APE communication network. The link ports help connects the APE string in a chip to another to form a longer string.

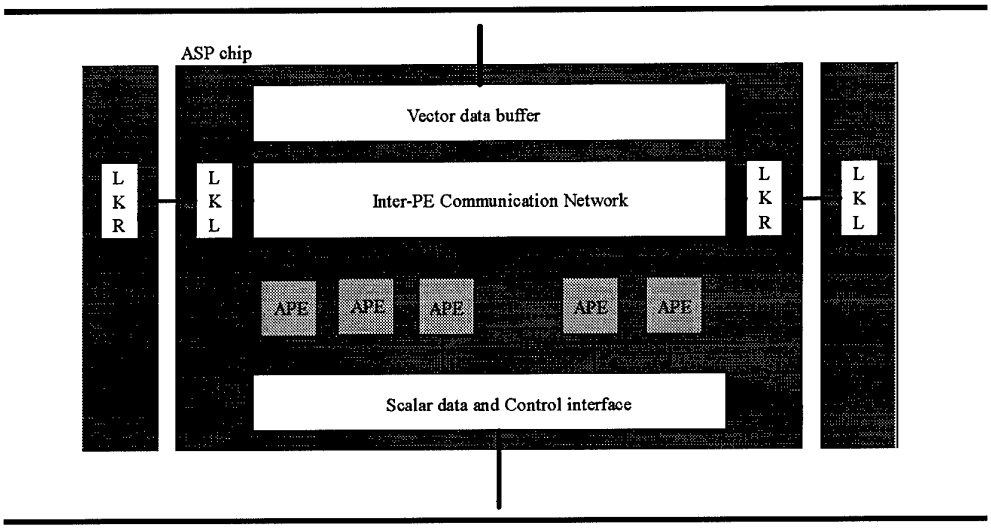

Fig 3.5 The ASP string

ASP uses content addressing rather than location addressing techniques. Thus, APEs are selected for subsequent parallel processing by comparing their data and activity contents with the states of the corresponding data and activity buses. In operation, the ASP supports a form of set processing in which the subset of active APEs (those which match broadcast data and activity values), support scalar-vector and vector-vector operations. The ASP either directly activates matching APEs or uses resources to indirectly activate other APEs. The match reply line indicates whether or not any APEs match. The controller either directly broadcasts scalar data or receives it via the bit parallel data bus.

Each substring may be partitioned into programmer defined segments, separated by segment links, in support of structured data such as arrays, trees, tables, graphs etc. The segment links can be opened or closed to prevent or allow the transmission of inter-APE communication signals between adjacent segments. Thus, each segment comprises a span of contiguous APE blocks, with internal block links closed and end block links converted to segment links. Users can create variable length segments by writing segment links corresponding to the M-tags at the ends of APE blocks. Alternatively, they can create equal length segments, comprising power-of-two APEs, with a special command.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

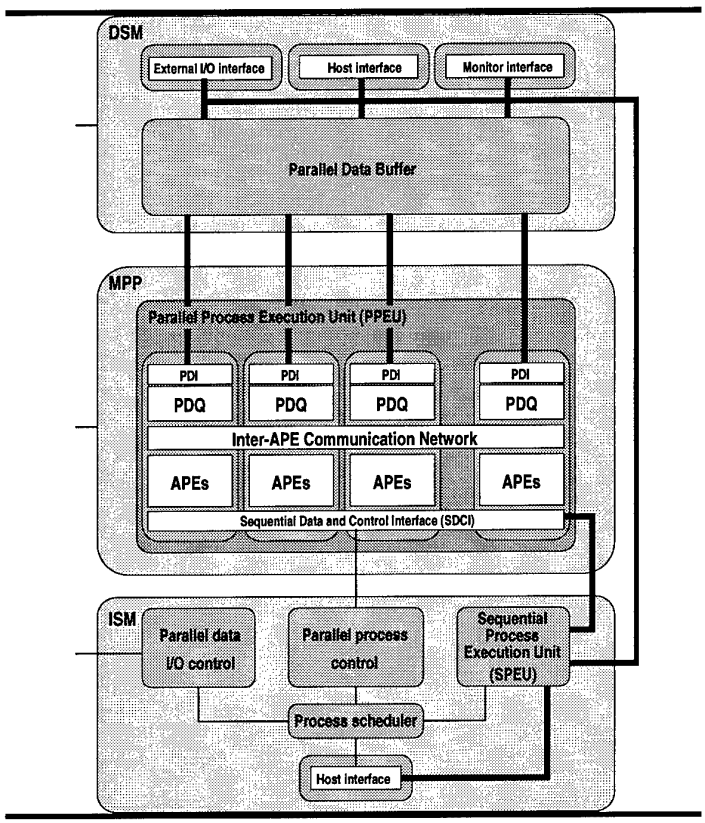

Fig 3.6 Modular-MPC ASP Modules

An ASP module (Fig 3.6) has a private Parallel Data Interface (PDI) for the transfer of parallel data. The Modular-MPC uses a hierarchy of parallel data pipelining to transfer parallel data form the interfaces to the outside (i.e. host interface, external I/O interface and monitor interface) to the Parallel Data Interfaces (PDIs) of the ASP modules and vice versa. The lowest level in this hierarchy comprises the Parallel Data Queue (PDQ). Data transfers between the PDQ and the ASP modules are called *Primary Data Transfers* (PDT). The PDQ is implemented with an orthogonal data queueing mechanism. Data is loaded, overlapped with parallel processing, word-sequentially, bit parallel. It can subsequently be exchanged with the APEs in a word-parallel bit-sequential manner. Due to the massive bandwidth of this exchange (e.g. with a bandwidth of 40 Mbits/sec for each APE a Modular-MPC comprising 64k APEs has a bandwidth for Primary Data Transfers (PDTs) of 2.63 Tera bits/sec), the exchange time during which parallel processing has to be stopped, and consequently overheads, can be reduced to a minimum.Due to its orthogonal structure, the PDQ scales linearly with the number of APEs in the string.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

ASP modules provide a simple means for scalability. Performance can be adjusted to the application requirements by changing the number of APEs per ASP module. I/O requirements, which manifest themselves in the required number of data channels, are met by changing the number of ASP modules.

3.3.3 The Data Stream Manager

The DSM contains a DSM processor, a host interface, and support for tertiary data transfers.

The DSM processor is responsible for controlling the secondary and tertiary data transfers. It is responsible for initialising the corresponding data controllers and also performs sequential operations. It appears as the bus master on the DSM internal bus, which allows it to access to all resources on the bus. The DSM processor has hardware support for synchronising the instruction stream manager as well as for interrupting a higher level controller via the host bus. The processor itself can be chosen according to the functionality required at this level e.g. a Sparc processor could be used for high functionality- high performance applications

The Data Stream Manager (DSM) is connected to the Massively Parallel Processor (MPP) via a multi-channel, large bandwidth parallel data bus.

The Data Stream Manager (DSM) consists of interfaces for parallel data I/O and the Parallel Data Buffer (PDB). The PDB, which supports image patching, is crucial to minimise delays caused by parallel data I/O. The Parallel Data Buffer (PDB) provides storage for at least one, but usually several, images, similar to a frame-store. Each Parallel Data Buffer (PDB) is connected to the PDBs of other Task Execution Units (TEUs) via the Inter-Task Parallel Data Router (ITPDR).

Parallel processing applications vary considerably in their I/O requirements. In order to achieve a high level of *machine versatility* the Data Stream Manager (DSM) is designed as a highly modular unit, which can adapt to the whole spectrum of I/O requirements. In particular, the DSM is designed to minimise I/O overheads through a successive increase in input bandwidth between the stages of a data pipeline and patching overheads by providing a mechanism for high-bandwidth patch exchanges.

Depending on the I/O requirements of a particular application, different data pipelines (see Fig 3.7) can be implemented.

1. 1-stage pipeline

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

In this configuration of the DSM the parallel data interfaces are directly connected to the ASP modules, parallel data is only buffered in the Primary Data Queue. Consequently any image storage is external. Two levels of parallel data transfers can be observed:

Secondary Data Transfer (SDT) : Data is transferred between the external interfaces and the Primary Data Queue (PDQ). This transfer is comparatively slow, since the PDQs of all ASP modules have to share a single connection to the interfaces. However, Secondary Data Transfers (SDT) can be overlapped with processing.

Primary Data Transfer (PDT) : This is the word-parallel transfer between the Primary Data Queue (PDQ) and the ASP modules It is a non-overlapped transfer with an extremely high I/O bandwidth.

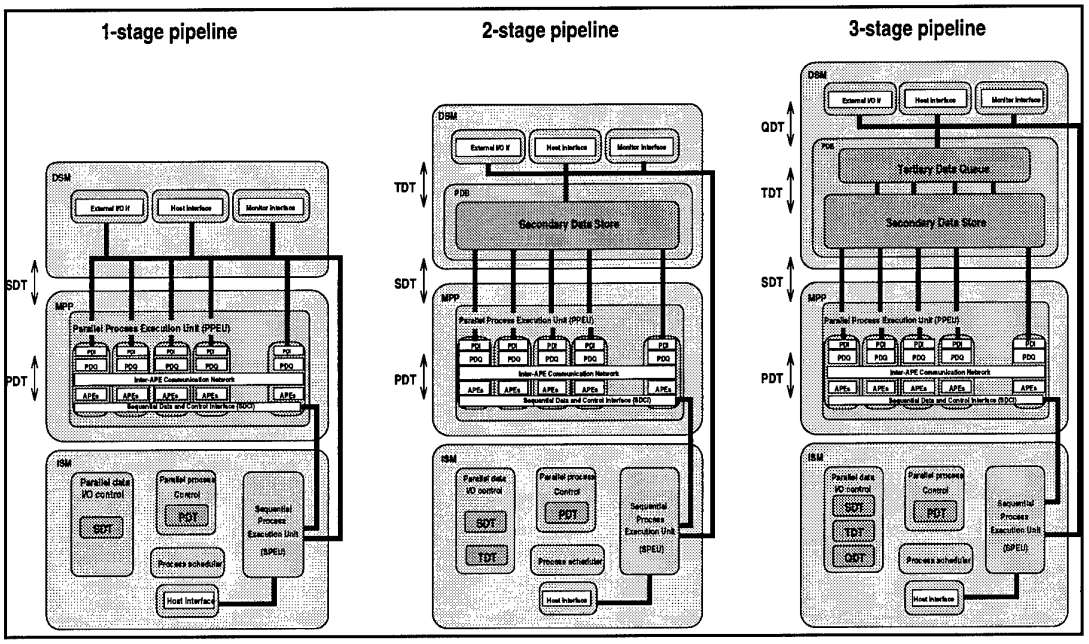

Fig 3.7 DSM configuration options

2. 2-stage pipeline

A 2-stage pipeline is configured by including the Secondary Data Store as a further stage in the parallel data I/O pipeline. In this configuration, patch processing is supported by the DSM. The Secondary Data Store (SDS) implements a multi-frame store, which is large enough to store at least one, but typically several images. By storing or loading of sub-images while the processing of the current sub-image takes place, the inefficiencies caused by patching are minimised. Frequent exchanges of subimages with the

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

MPP are made possible by a high bandwidth parallel data I/O. Compared to the 1-stage pipeline a further data transfer has been introduced:

Tertiary Data Transfer (TDT) : This transfer takes place between the external interfaces and the Secondary Data Store (SDS) via a single data channel for image transfer.

Secondary Data Transfer (SDT) : Other than for the 1-stage pipeline, this transfer now takes place between the Secondary Data Store (SDS) and the Primary Data Queue (PDQ). It implements a fast, multi-channel patch transfer which is overlapped with processing, thus minimising overheads. The number of data channels for the Secondary Data Transfer (SDT) is chosen to meet the application requirements.

Primary Data Transfer (PDT) : The PDT is implemented in the same way as described for the 1-stage pipeline.

3. 3-stage pipeline

With the introduction of a Tertiary Data Queue (TDQ), the DSM can be configured as a 3-stage pipeline. Depending on the I/O requirements of the application and typical memories, a performance advantage can be achieved by including the Tertiary Data Queue (TDQ) in the parallel I/O data pipeline. Thus, it is used for the cost-effective minimisation of I/O overheads and as an intermediate storage to buffer data between the interfaces and the Secondary Data Store (SDS). Four levels of parallel data transfers can be observed for the 3-stage pipeline.

Quaternary Data Transfer (QDT) : This transfer takes place between the Tertiary Data Queue (TDQ) and the interfaces via a single data channel.

Tertiary Data Transfer (TDT) : In contrast to the 2-stage pipeline, for this configuration the TDT takes place between the Tertiary Data Queue (TDQ) and the Secondary Data Store (SDS). The number of connections for this transfer can be configured according to application needs.

Secondary Data Transfer (SDT) and Primary Data Transfer (PDT) : As compared to the 2-stage pipeline, these transfers remain unchanged.

The Secondary Data Store (SDS) provides storage for one or more image frames for fast patch processing. Therefore, it needs to be scalable according to the storage requirements of the application, independently of

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

the number of Associative Processing Elements (APEs) and independently of the number of I/O data channels for between the SDS and the MPP. Fig 3.8 shows the modes in which the SDS can be configured.

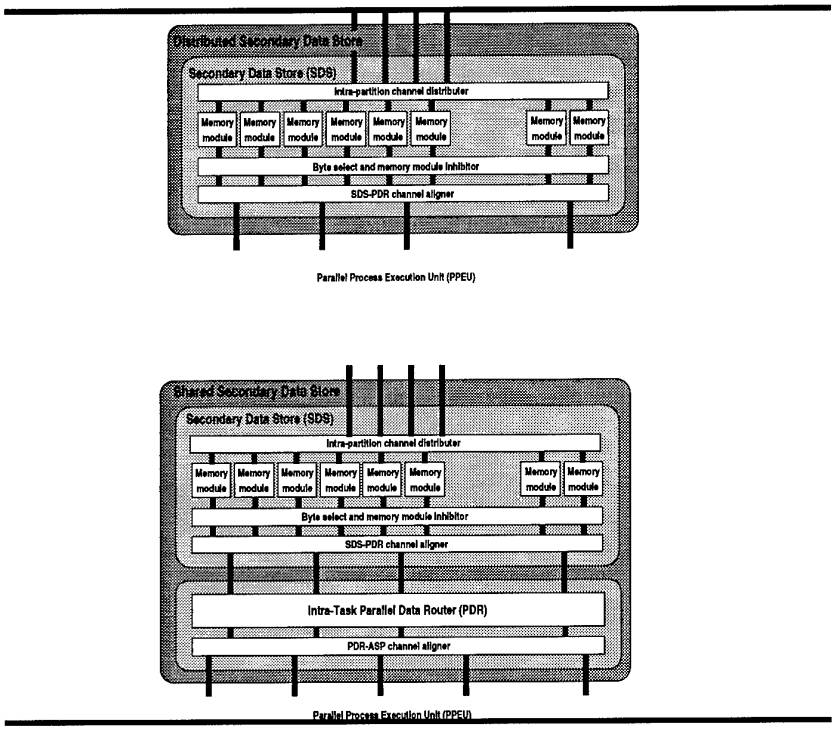

Figure 3.8 shows the two modes SDS configurations

Distributed Secondary Data Store (SDS) :This mode implements static routing between the different blocks of the SDS memory and the ASP modules. It is used for those applications where ASP modules access data only in assigned memory modules

Shared Secondary Data Store (SDS) : For a number of applications, ASP modules may access data in any memory module. In this case the SDS can be implemented in a shared mode with dynamic routing between memory modules and ASP modules. This dynamic routing is implemented with the Parallel Data Router. Consequently, through the flexible allocation of memory, the *application flexibility* of the Modular-MPC is increased significantly. A programmer sees the shared SDS as a single (shared) memory.

The Secondary Data Store memory implemented as a set of memory modules. Each module holds a fraction of the data kept in the SDS and is assigned its private I/O channels. The storage capacity of a memory module can be selected from a range of sizes. The SDS memory provides two modes of data filtering,

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

implemented by the byte select and memory module inhibitor. With byte select, any set of bytes in a storage word can be masked for write access such that only part of the memory word is updated. Byte masking is common to all memory modules. Write access to any memory module can be inhibited. The inhibition of a memory module can be data-dependent, i.e. only data with a specific signature is written to the SDS. The inhibition of one memory module is independent of all other memory modules.

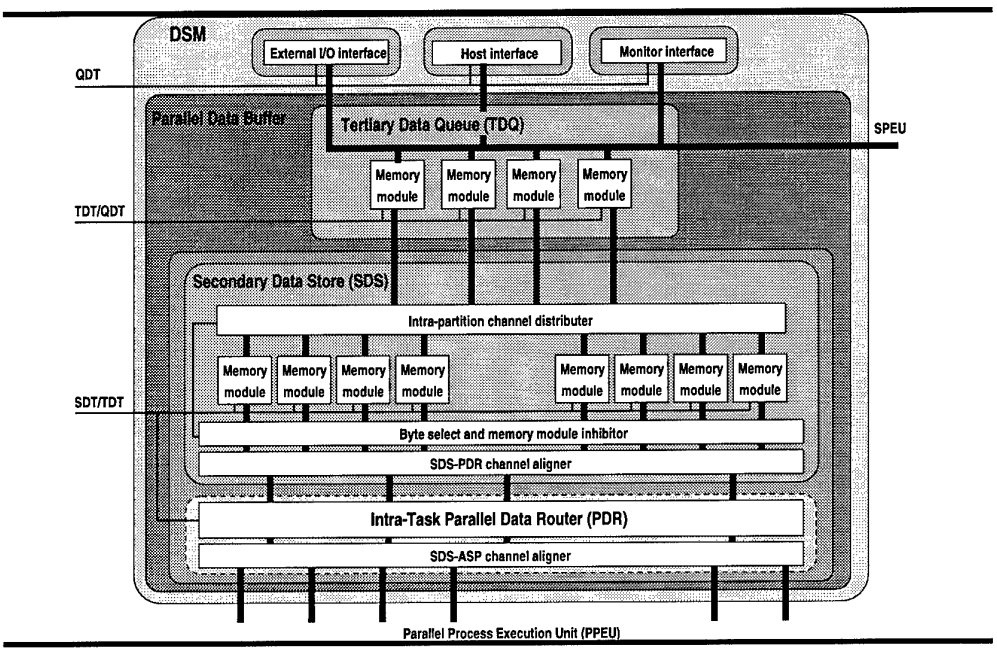

Fig 3.9 Tertiary data queue (TDQ)

The Parallel Data Router (PDR) implements the dynamic routing required to implement a shared SDS. Therefore, it should not be seen as a general purpose network, but as a router. The PDR implements a cross-bar topology, i.e. any ASP module can exchange data with any memory module. The number of data channels in the PDR can be adjusted according to application needs.

Similar to the Secondary Data Store (SDS), the TDQ (see Fig 3.9) is implemented as a set of memory modules. As for the SDS, each memory module holds only a fraction of the data, has its private I/O channel and its capacity can be selected from a range of sizes.

3.3.4 The Instruction Stream Manager

The Instruction Stream Manager (ISM) consists of the host interface, the process scheduler, (which schedules the parallel data I/O control and the parallel process control) and a Sequential Process Execution

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Unit (SPEU). A single channel control connection joins the parallel process control with the MPP (SIMD concept).

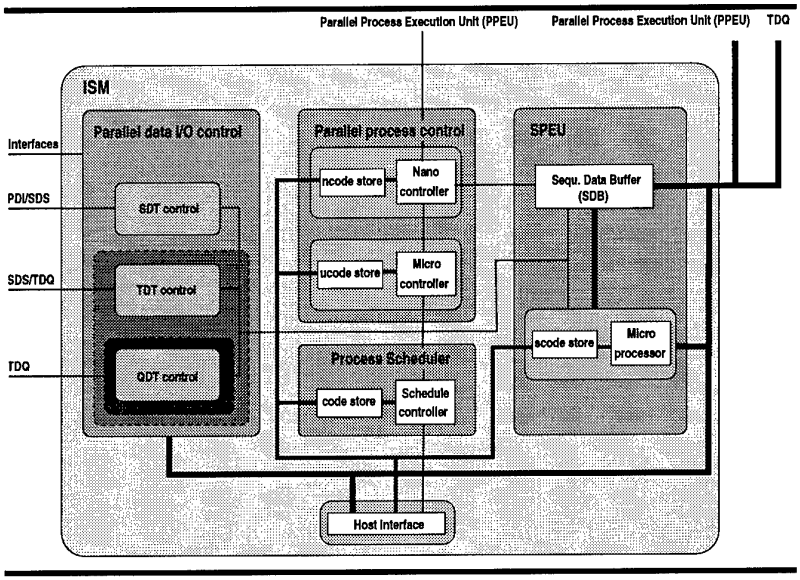

Fig 3.10 The Instruction Stream Manager (ISM)

Figure 3.10 presents an overview over the main functional blocks in the Instruction Steam Manager (ISM).

The process scheduler controls and 'orchestrates' all other units in the ISM. Through remote calls it starts parallel processes in the parallel process control, sequential processes in the Sequential Processes Execution Unit (SPEU) and it initiates data transfers controlled by the parallel data I/O control. Furthermore, the process scheduler uses the feedback from these units to align further scheduling of processes. As the main controlling unit in the ISM it is the only functional block which is directly connected to the host workstation via the host interface. The process scheduler is assigned a private code-store, which can be loaded via the host interface.

The parallel process control generates the control stream for the Massively Parallel Processor (MPP). As a cost-effective way to minimise control overheads, it is organised as a control hierarchy.

The *micro controller* calls blocks of operations which are remotely executed in the *nano controller*. While the nano controller, which is specifically designed for a fast repetition of blocks of operations, executes the remote call, enough time is created for the micro controller to organise the "housekeeping" (e.g. branching) of the parallel process control stream. Furthermore, the nano controller provides functionality to concatenate sequential data, which is stored in the Sequential Data Buffer, and control to be sent as a single instruction to

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

the MPP. Both, micro controller and nano controller have private code stores in which the required *task* and *processes* software library modules are downloaded via the host interface, prior to processing.

The Sequential Process Execution Unit (SPEU) is dedicated to the execution of sequential *processes*. Thus it plays a crucial role in minimising overheads due to operations with minimal natural parallelism.To this end, the SPEU includes a fast floating-point microprocessor with a dedicated code-store for sequential process code which can is downloaded via the host interface.

Furthermore, the SPEU includes a data store dedicated to sequential data, called Sequential Data Buffer (SDB). The SDB stores three kinds of sequential data intermediate data used as sequential data in the MPP, sequential data being evolved by the microprocessor of the SPEU and parameters for the parallel data I/O control.

Furthermore, the SDB acts as a bridge to move parallel Data from the Data Stream Manager to the Instruction Stream Manager (ISM) where it can be used as sequential data for further processing.

The parallel data I/O control governs data transfers between all levels in the Data Stream Manager . Depending on the number of stages in the Parallel Data Store, the parallel data I/O control implements a hierarchy of modules (see Figure 3.9):

1-stage pipeline only the Secondary Data Transfer (SDT) control is required

2-stage pipeline SDT control and TDT control have to be implemented

3-stage pipeline all levels of control, SDT control, TDT control and QDT control are required

Figure 3.11 demonstrates how the functional units in the parallel data I/O control can be partitioned in two sets:

-coordinating units : These include SDT control, TDT control and QDT control. They coordinate transfers between units in the DSM by controlling address generating units.

-address generating units : Although the rather large number of address generating units might seem confusing at first glance, these units follow a simple architectural principle. Each functional unit in the DSM parallel data I/O pipeline is assigned a private controller together with, if necessary, an address generator as follows:

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

the external interfaces are controlled by the interface control

the Tertiary Data Queue is controlled by the TDQ address generator, which also provides the addresses for each memory module in the TDQ

each memory module in the Secondary Data Store (SDS) is controlled and provided with addresses by a local address generator

the Parallel Data Router (PDR) is configured and controlled by the PDR configuration unit

the Parallel Data Interfaces (PDI) are controlled by the PDI control and the ASP channel inhibitor

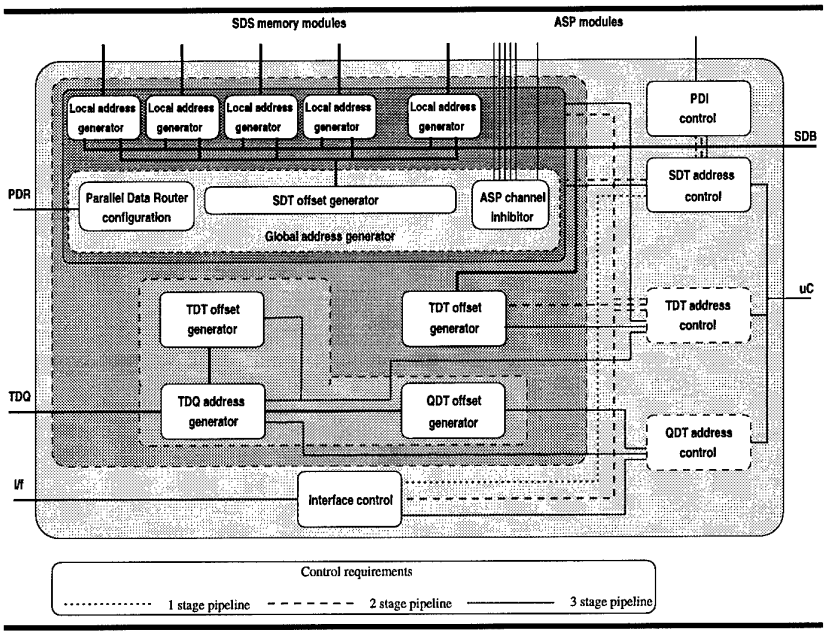

Fig 3.11 I/O control

The Secondary Data Store (SDS) and the Tertiary Data Queue (TDQ) are accessed by two transfers, SDT and TDT, and TDT and QDT respectively. Therefore, their assigned address generators are able to switch between two different contexts, depending on the transfer which is executed at a given time.

The offset generation for the local address generators is done in a similar way as the one described for the TDQ address generation. However, additional functionality is included for the case of a shared SDS when the Parallel Data Router is included in the SDS. In this configuration, for Secondary Data Transfers, offsets have to be generated for each local address generator, the parallel data router has to be configured and ASP channels have to be inhibited in cases of access contention to certain memory modules in the SDS. All this is

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

handled by the global address generator, which executes these tasks overlapped with the actual address generation for SDT in the local address generators. Parameters for the different address generators can also be downloaded from the Sequential Data Buffer (SDB).

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

4

NEO detection synthesis

This section describes the synthesis of the NEO detection application using a modular massively parallel computer currently being developed at Aspex Microsystem Ltd. The NEO detection implementation has been studied at two levels of abstraction namely, the task and the program level to increase overall detection performance and efficiency.

4.1 Synthesis approach

The approach adopted to implement the NEO detection program aims to maximise overall PE utilisation and efficiency at both the task level (pixel : PE ratio) and at the program level (effective PE usage during processing task sequences). At the task level, PE utilisation is dependent on the pixel : PE ratio utilised (or the size of the virtual PE) for a sequence of tasks without requiring any I/O. While maximum task efficiency is ensured by reducing the overall overheads, especially due to communications. Most often, the choice of a particular task implementation is usually a compromise between PE utilisation and task level efficiency. Although a task might appear to be running efficiently, PE utilisation may be low and would have a bearing on the overall program efficiency. Effective PE count would reduce and hence active patch sizes processed by the array reduces.

Once tasks have been developed, the sequencing and partitioning of tasks is considered for overall program synthesis. During the course of the processing, data transforms from its iconic 2D data structure through to a collection of object-based iconic structures and finally into a symbolic description for each object, as illustrated in Fig. 4.1.

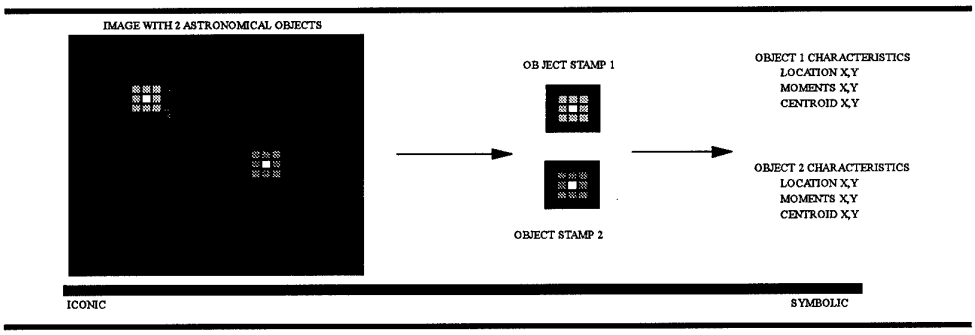

Fig. 4.1 Evolution of data during processing

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

The implementation of the NEO detection program is based on the exploitation of the massive data volume reduction ensuing low and intermediate level image processing operation, typical in astronomical image processing. Also, exploitation of these evolving data structures can reduce communication overheads incurred during task processing.

At the program level, PE utilisation is seen as the ratio of active PEs to the total number PEs. If the implementation does not match evolving data structures, the under-utilisation of available resources results .

Given this evolving data structures and the massive data reductions (95-100%), the tasks have been partitioned into two main group. Those tasks responsible for reducing images to objects generally have to be able to keep up with the data rate. These tasks namely image restoration, object enhancement and segmentation, are involved in time-dependent processing (TDP). Exploiting this reduction from pixel-based to object-based, the subsequent tasks namely the analysis and decision tasks are object-dependent (ODP). Additionally, analysis is done on object stamps and are hence iconic-based while decision involves symbolic processing. This partition can be seen in Fig 4.2

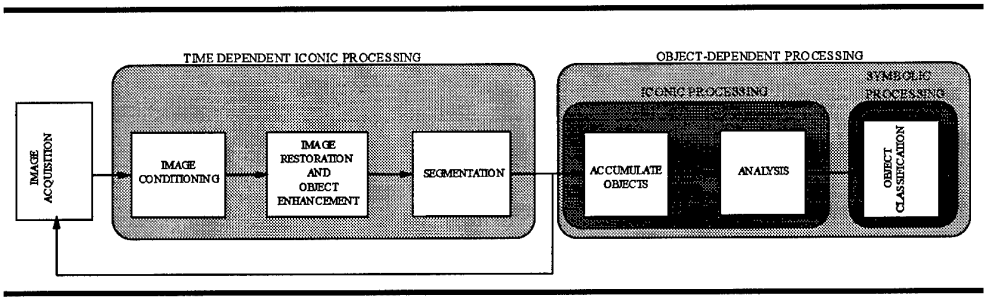

Fig 4.2 Task partitioning : time-dependent and object-dependent processing

4.2 NEO application-specific synthesis

The NEO detection simulation is based on the GODS[5] program [SAV94] developed at O.C.A. However, as the decision stage is the least complex and computation intensive of all the tasks, the consequent modelling of system performance, based on the Modular-MPC system, was performed on the more intensive TDP and ODP iconic-based stages.

4.2.1 TDP implementation

Raw images acquired from the CCDs are loaded into PEs with 1 pixel/PE. Each pixel is 16 bits in length. The TDP program from is shown in Fig.4.3.

[5]GODS : Global Orbit Determination System

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

TDP tasks include the "Mexican Hat" filtering, thresholding and maxima detection. The structure of the convolution mask that represents the "Mexican Hat " filter is octa-symmetric as seen in Fig 4.4. Typically performance increases may be achieved by exploiting this symmetry to reduce overall number of multiplication from $n \times n$ to approximately $n \times n$ DIV 8. Also given the nature of the gaussian-based filter, separable filters may also be employed. Separable filters reduces overall computations from $n \times n$ to about $4n$.

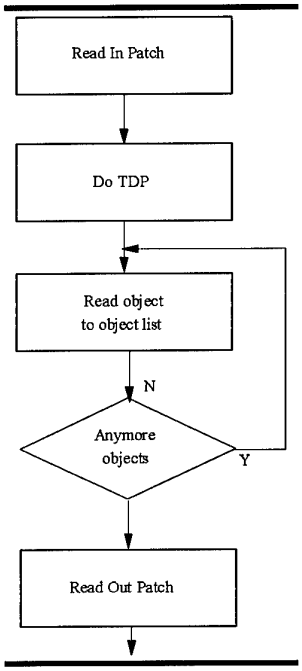

Fig 4.3 Flow diagram for Time Dependent Processing

However, studies of several convolution techniques [JOS94b] for a string-connected architecture has revealed that minimisation of multiplication and the use of separable filters do not necessarily result in an increased performance. The most suitable technique is a compromise between the computation, intermediate result storage and the communication required between PEs. This is due to the fact that PE utilisation decreases with the need for more intermediate storage facilities, resulting in larger communication overheads due to increase virtual PE sizes and hence reducing task efficiency.

The implementation techniques adopted is based on a method that exploits the string-connected architecture of the ASP. This method is accomplished in two major steps to complete the multiply accumulation on either halves of the mask. Raw data is first copied into free memory within each PE. This establishes an image copy that can be moved by shifts equal to the distance of the next involved PE along the string. Immediate neighbours, east or west, depending on which step is presently done, are first communicated for convolution. As this is moved, pixels following it ride on the transfer after which it is stored again. Hence this method is called piggy-backing. The total communication is directly related to only the diameter of the convolution mask on the string. The result of the multiply and accumulate is stored in the result space. Hence no additional memory is required for intermediate results.

a	b	c	d	e	d	c	b	a
b	f	g	h	i	h	g	f	b
c	g	j	k	l	k	j	g	c
d	h	k	m	n	m	k	h	d
e	i	l	n	o	n	l	i	e
d	h	k	m	n	m	k	h	d
c	g	j	k	l	k	j	g	c
b	f	g	h	n	h	g	f	b
a	b	c	d	e	d	c	b	a

Fig.4.4 Symmetricity of the "Mexican Hat" Filter

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Thresholding is achieved by a scalar-vector comparison. Pixels found below the threshold are set to zero. After thresholding, objects are defined by their point of maxima. Maxima points are determined by comparison with their 4-way connected nearest neighbours. Maxima detection begins by considering all pixels belonging to a set of possible objects. As the algorithm proceeds, a maxima set is built by rejecting all objects that have at least one neighbour greater then itself.

Before objects can be read-out, x-y co-ordinates are generated per patch. Instead of using a *log n* method for this an algorithm that uses an external address-based look up table is used. Using the ASP's activation modes, only patch width and patch height reads and writes are necessary.

Fig 4.5 shows the utilisation of the PE during the processing.

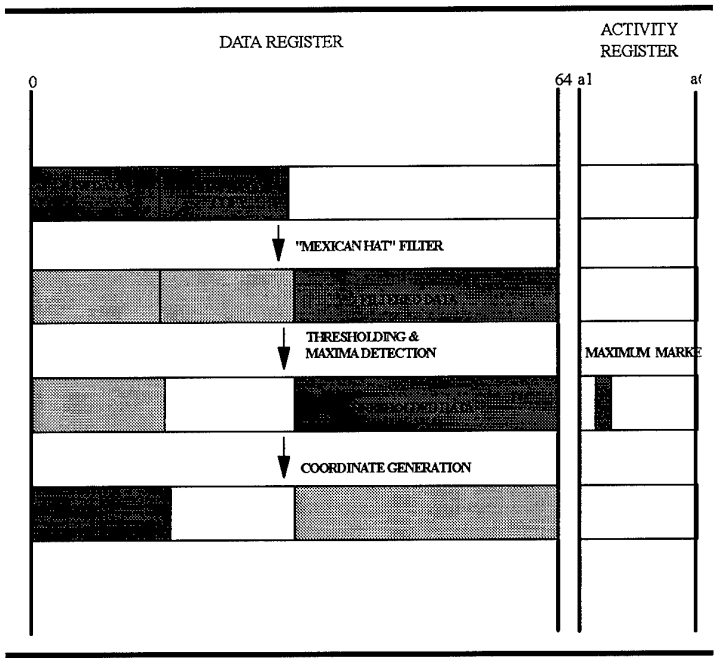

Fig 4.5 Memory utilisation per APE during TDP

At this point in processing, the overall 2-D image structure has been reduced to a collection of points representing astronomical objects. Before objects may be read out, information regarding their location in terms of pixel co-ordinates have to be generated and associated with them. Pixels are now 32 bits long for at least 3 decimal place accuracy while co-ordinate length is dependent on the patch size. The (x,y) pixel co-ordinates of the maxima points are read out through scalar reads via the Sequential Data Interface. Since the number of objects expected to be detected per patch, it may be more efficient to attempt scalar reads rather than transfer co-ordinates through the parallel data buffer (PDB). The processed patch is also read out through the PDB.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

4.2.2 ODP Implementation

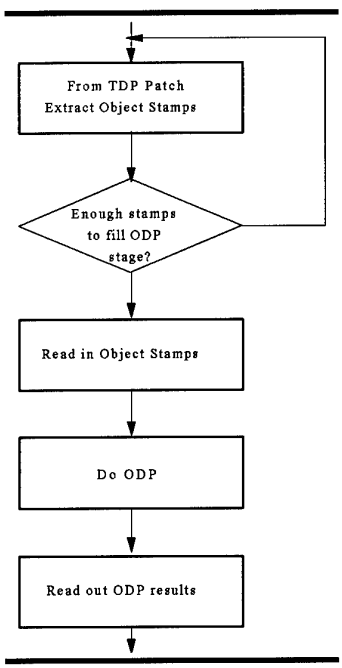

Fig. 4.6 Flow diagram for Object Dependent Processing

The ODP program is shown in Fig 4.6. For the analysis, stamps extracted from the processed image around detected objects centred on maxima co-ordinates, are loaded into the array. Each stamp occupies ($nxn+5$) PEs as shown in fig 4.7. The 5 PE cell pack consist of one intermediate accumulator cell, Moments X and Y cells and the Centroid X and Y cells. This cell pack neighbours the central PE of the stamp. This is in order that communication overheads can be minimised. This scheme provides sufficient memory for any intermediate results and also the huge results of the moments. To avoid dynamic segmentations, processing is tightly coupled to active sets only, thus preventing unnecessary writes of intermediate results in inactive cells. Also, by allocating necessary memory for each result, excessive overheads due to clearing PEs to make space for centroid calculations can be avoided

The analysis begins with the moments calculation. The moments mask is bi-symmetric as seen in Fig 4.8. The technique employed reduces multiplication by accumulating all pixels having the same multiplier. Again by using the piggy-back method, communication overheads can be reduced directly. This method is useful since multiplication involves 45 bit long results. This is again so that precision of 3 decimal places are maintained.

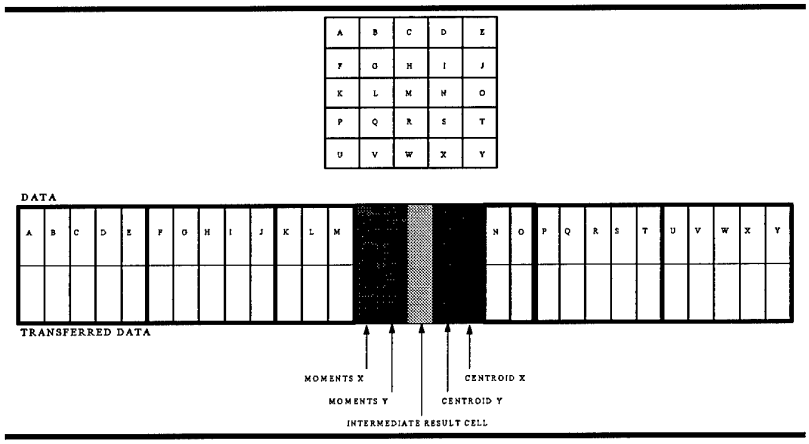

Fig 4.7 Mapping moments on to a string

Centroid calculations involves only the determination of the numerator and denominator values (a and b) so that division can be accomplished outside the ASP. External division calculations are less costly.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

Besides even for the decision stage, centroid values are only required for astronomical co-ordinate reduction rather than the detection of asteroid motion.

Only the results of the moments and centroids are read-out. These are used to build a list of records of detected objects. List corresponding to the different frames of the same field are then used for the decision process. Given the huge data reduction available from the TDP stage and the sparse, random nature of astronomical object distribution, there would be periods when ODP is not run due to patches in the TDP without any object and/or there is insufficient data to fill the array. During these times, decision tasks may be executed.

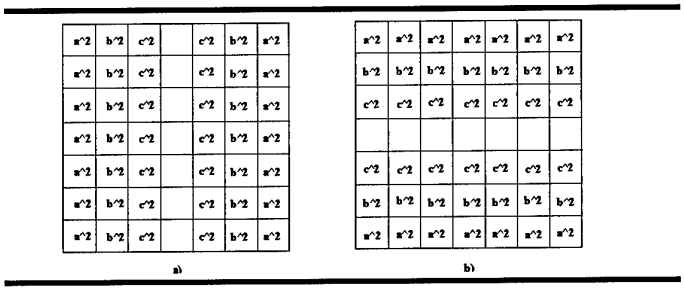

Fig 4.8 Symmetry in moments masks a) X Moments and b) Y Moments

4.2.3 NEO detection program implementation

The TDP stage accepts raw data from the CCD controllers via the external I/O interface. CCDs are read line be line and all CCDs are read simultaneously, for example a 9 2k x 2k CCD column would give a burst of 2k x 9 pixels per read-out. Incoming CCD data is doubled buffered. Patches are extracted from the buffer and are processed Note the a circular buffering scheme should be employed to maintain scan-continuity.

Once processed, the TDP passes the processed patches, instead of object stamps, and the list of detected objects to the second stage. The choice of passing processing patches over object stamps from the TDP to the ODP stage, is so as to maintain some level of determinacy in the data size and hence the communication time. Object stamps introduces data duplication between stamps, as star regions may overlap. Hence, if the patch contains a star cluster, many objects may discovered resulting in "data explosion". The object co-ordinates obtained from the TDP are patch co-ordinates. These co-ordinates are converted to absolute co-ordinates before being passed to the ODP. The overall program requirements are shown in the data and control flow diagram in Fig 4.9.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

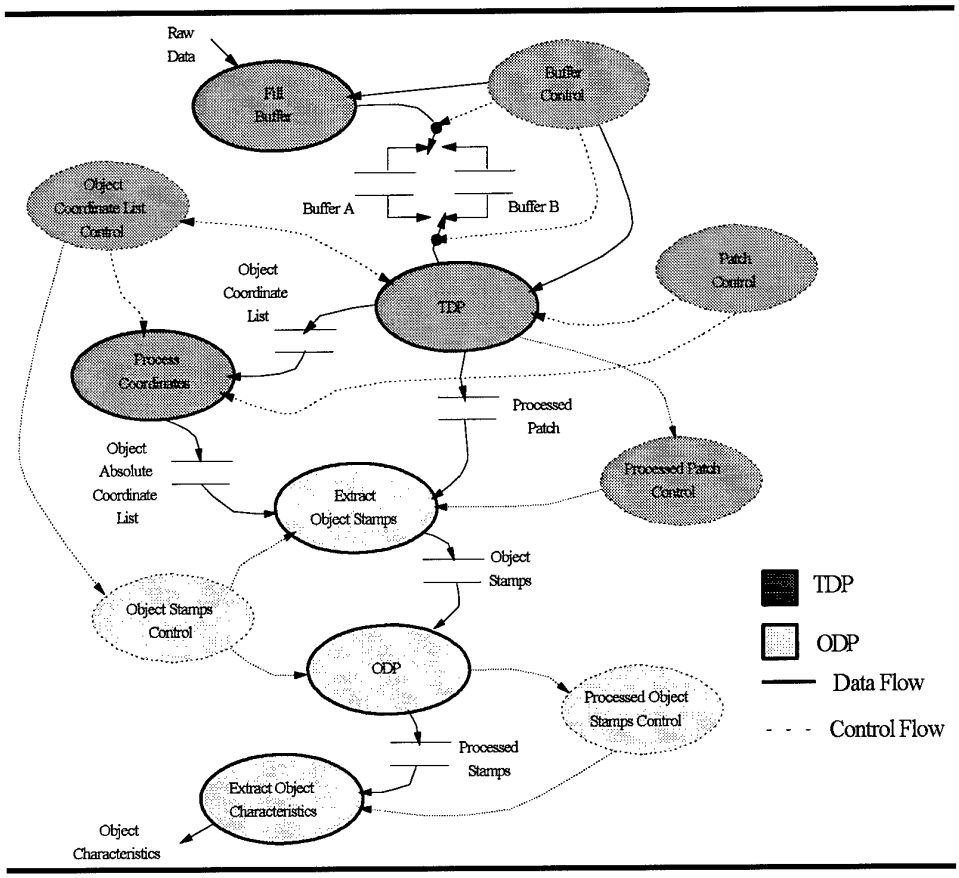

Fig 4.9 Data flow and control flow diagram for the NEO detection program

4.2.4 Modular-MPC system implementation for NEO detection

Two approaches have been studied for the overall system; a 2-stage and a 1-stage processing pipeline seen in the overall functional block diagram in Fig. 4.10. Both approaches are effectively data driven, i.e. the TDP begins processing once the buffer is filled while the ODP begins when enough object stamps have been accumulated.

In the 2-stage pipeline where the TDP and the ODP stages are spatially-partitioned (Fig 4.10a) and operate asynchronously. The 1-stage processing pipeline (Fig. 4.10b) multiplexes between the TDP and the ODP stages i.e. temporally partitioned i.e. they operate on the same stage. With this approach, TDP is repeated until enough object stamps are accumulated before the ODP. The number of repetitions necessary is dependent on the expected object detection rate and hence the viewing depth for which the imaging unit has been configured for.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

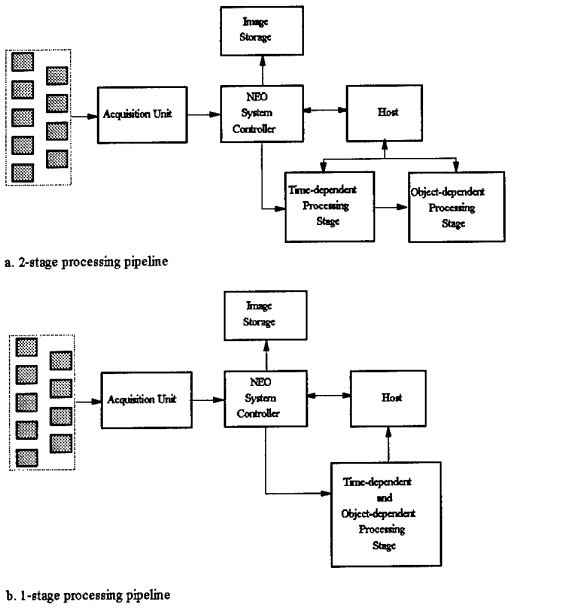

Fig 4.10 NEO detection system with TDP and ODP processing stages

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

5

NEO Simulation & evaluation

The simulation for the NEO detection program was done on the ASP System Testbed for Research and Application (ASTRA). The results of this simulation could then be applied into the appropriate system model with the task partition mentioned in the previous section. The ASTRA system was developed by Aspex Microsystems Ltd, to develop algorithms and assess the ASP in a wide range of off-line signal and data processing applications.

ASTRA can be represented functionally as comprising four blocks : the high level controller, intermediate-level controller, the low-level controller and the processing array, as seen in Fig 5.1. It. is based on the traditional 64 PE VLSI implementation of the ASP string. The testbed can accomodate upto 8K processors, with 2K processors on a 9U hyper-extended Eurocards. The host for the ASTRA is a Sun workstation and provide overall program control, external data storage and user interface facities. A intermediate controller based on a Motorola 68030 handles ASP procedure sequencing and also handles the array I/O.The array is controlled by a AMD 29330 micorsequencer and microprogram storage. A second generation ASTRA is currently being developed that could utilise ether the 64-bit VLSI or the hybrid-WSI implementations. Each board would have about 8K processors using the hybrid WSI chips and the host would be a Sunsparc station.

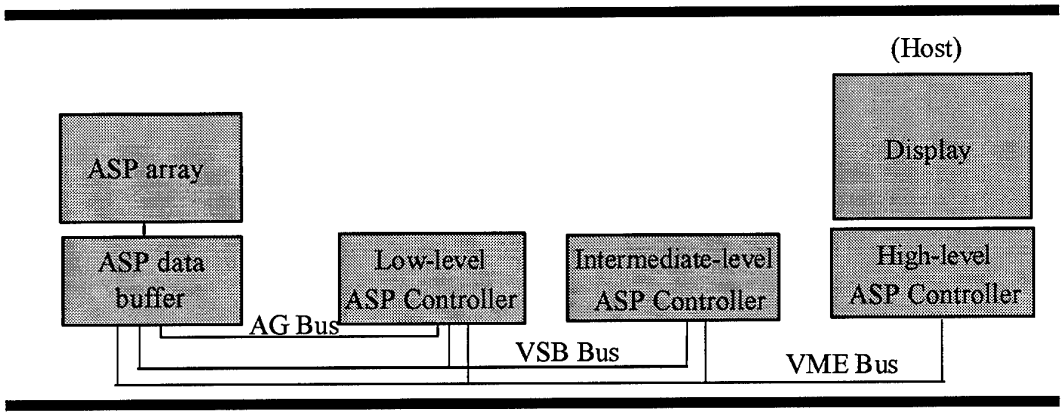

Fig 5.1 The ASTRA system

The main objective of the simulation were to obtain a realistic assessment of the ASP-based tasks within the system's hardware and software contraints. Image restoration and enhancement, segmentation and analysis algorithms were written and developed on ASTRA. As only minimal instruction changes would

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

exist between the VLSI ASP implementation and later generation ASP chip, algorithms and hence, total algorithm slot time would remain similar.

The program implementation of the pixel-based and iconic based tasks was sequential in nature since only a single instruction thread is possible on the ASTRA. With the functionality provided by the ASP compiler, list files maybe generated revealing the macro-to-micro instruction break down. This list is useful in determining overall time slots required per array procedure.

The Table 5.1 below gives the execution times of the different TDP and ODP iconic-based tasks for varying segment widths and operator sizes.

Algorithm	Time (μsec)	
	Object Size : 5	Object Size : 15
"Mexican Hat" Filter[1]	1,178 - 1,357	3,578 - 4,205
Threshold	1.9	
Maxima Detection[1]	57 - 236	59 - 238
Moments X	545	1,679
Moments Y	545	1,679
Centroid X & Y	77	85

[1]Segment width from 32-256 PEs

Table 5.1 Timings for the TDP and ODP tasks

The throughput for the TDP and ODP iconic-based stages are given below in Table 5.2. for an object size of 5-15 pixels (equivalent to operator size). The tasks were modelled to reflect their relation to the patch width, the number of processing elements, the size of the operators, the number of CCDs and other system variations such as the number of I/O channels.

#PEs	TDP Throughput (10^6pixels/sec)		ODP Throughput (Objects/sec)	
	MaskWidth : 5	MaskWidth : 15	MaskWidth : 5	MaskWidth : 15
4K	2.94	1.03	116,894	5,174
8K	5.89	2.06	233,790	10,348
16K	11.78	4.11	467,680	20,696
32K	23.56	8.22	935,160	41,392
64K	47.11	16.45	1,870,320	82,783

Table 5.2 Throughput for the TDP and ODP stages

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

The large variation in timings for the TDP stage for increasing PEs is due to, not only the coordinate generation, but also mainly because of the scalar read required for object coordinates and is hence dependent on the number of objects detected. The throughput of the ODP could be increased even more by compromising communication overheads with a more compactly packed object stamp on the ASP string.

With the above figures for the different tasks,, appropriate system models were then designed to evaluate the detection program on the Modular-MPC. Factors taken into consideration in the models are mainly related to processing and data movement. Table 5.3 and Table 5.4 gives possible configurations for a 1, 2 or 3 CCD column (using Kodak 2k x 2k CCDs) system at varying acquisition rates (sidereal rates) and smallest and largest expected object sizes for the 1 stage and 2 stage pipeline implementations. The larger configurations use the multi-channel input to reduce the I/O overheads. Thus, from the table below, a 16K system would produce 14 objects after the TDP stage. And if the system had 16 Kodak 2K x 2K CCDs in a single column then it would be able to objects of size less than or equal to 15 and 5 at sidereal rates of less than or equal to 3 and 10 respectively.

TDP		Columns						ODP
		1		2		3		
Number of PEs	Objects	(O_{max},S_{min})	(O_{min},S_{max})	(O_{max},S_{min})	(O_{min},S_{max})	(O_{max},S_{min})	(O_{min},S_{max})	Number of PEs
16K	14	(15,3)	(5,10)	(15,1)	(5,6)	(15,1)	(5,4)	4K
24K	20	(15,6)	(9,10)	(15,3)	(5,9)	(15,2)	(5,6)	5K
32K	27	(15,8)	(11,10)	(15,4)	(5,10)	(15,2)	(5,7)	7K
40K	30	(15,10)	(15,10)	(15,5)	(7,10)	(15,3)	(5,7)	8K
76K	33	(15,10)	(15,10)	(15,10)	(15,10)	(15,6)	(15,7)	15K

Table 5.3 Performance of various PE configurations for a given object size(O) and sidereal rate (S) represented as (O,S). The above representations shows the maximum object size (O_{max}) possible with the minimum sidereal rate (S_{min}) and the minimum object size(O_{min}) and the maximum sidereal rate (S_{max}) for a 2-stage pipeline.

The number of PEs for the object-dependent processing (ODP) stage was determined by the average number of objects expected. The Spaceguard Report had recommended an approximate object detection density of 30,000 objects/sq.deg. Thus given the machine size, the CCD angular resolution and this dnesity the average number of objects per patch is calculated. Based on this object rate and the number of PEs required to hold an object, the ODP size can be determined. The number of objects detected per patch is random in nature but for the purpose of the simulation this approximation would suffice.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

		Columns					
		1		2		3	
Number of PEs	Objects	(O_{max},S_{min})	(O_{min},S_{max})	(O_{max},S_{min})	(O_{min},S_{max})	(O_{max},S_{min})	(O_{min},S_{max})
16K	14	(15,3)	(5,10)	(15,1)	(5,5)	(15,1)	(5,3)
24K	20	(15,6)	(9,10)	(15,3)	(5,9)	(15,2)	(5,6)
32K	27	(15,8)	(11,10)	(15,4)	(5,10)	(15,2)	(5,8)
40K	33	(15,10)	(15,10)	(15,5)	(7,10)	(15,3)	(5,10)
48K	40	(15,10)	(15,10)	(15,6)	(9,10)	(15,4)	(5,10)
88K	86	(15,10)	(15,10)	(15,10)	(15,10)	(15,7)	(9,10)
128 K	108	(15,10)	(15,10)	(15,10)	(15,10)	(15,10)	(15,10)

Table 5.4 Performance of various PE configurations for a given object size(O) and sidereal rate (S) represented as (O,S). The above representations shows the maximum object size (O_{max}) possible with the minimum sidereal rate (S_{min}) and the minimum object size(O_{min}) and the maximum sidereal rate (S_{max}) for a 1-stage pipeline

It can be seen that the 1-stage pipelined system is comparable if not better than the 2-stage pipelined system for a similar count of PEs. This can be attributed to the fact that in the larger effective array for the 1-stage system reduces considerably the overheads due to necessary overlaps during patching.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

6

Results and Conclusion

The Spaceguard report has provided a benchmark from which detection programs can build on. Covering in excess of 6000 to 10000 sq. deg./month and at a limiting magnitude of around 24, discovery completeness could be maintained with an estimated 200-300 times the increase in computational requirements offered at Spacewatch. However, astronomers will continue to drive their instruments to cover larger areas using new techniques and devices such as optically multiplexed arrays[CRA94] to remove inter-CCD dead space, charge injection devices allowing non-destructive individual pixel read-out[NIN94], photon intensifiers to improve signal-to-noise ratios[FOR94] etc., to increase their individual observational efficiencies[LES94]. And as sensors become increasingly advanced and algorithms become more complex, computation requirements would indeed soar.

The approach that was adopted in this initial study for the feasibility of ASP technology to NEO detection has been straight forward. Two processing partitions have been considered namely a Time Dependent Processing stage that reduces images to extract objects and an Object Dependent Processing stage that handles all processing related to objects. By matching implementation to such a task sequence partition, overall solution performance is enhanced and effective tuning to specific system changes and upgrades is possible. The TDP stage can then be scaled to meet changes in image size (e.g. through larger CCD images in size and number), changes in the processing time window (e.g. decreasing acquisition time using more advanced sensors) and increased TDP task complexity. The ODP, on the other hand would be dependent on the number of objects that are expected per square degree and the resulting information extraction requirements.

It was seen in the previous chapter that the 1-stage pipelined implementation proved to require as many number of processing elements as the 2-stage. Having only 1 controller, in the 1-stage implementation, it proves to be more cost-effective than the 2-stage implementation. The performance obtained by various sizes of the machine with regard to the number of CCDs and

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

the acquisition time per CCD is given in Fig 6.1. A more realistic implementation, where in most cases, the object size would be about 9 pixels, Fig 6.2 demonstrates the resulting performance.

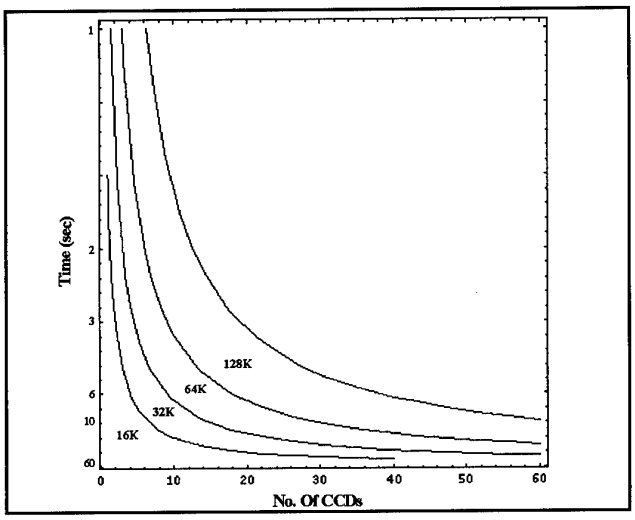

Fig 6.1. Performance of various configurations of the Modular-MPC with regard to the number of CCDs and the acqusition time for objects of size 15 pixels.

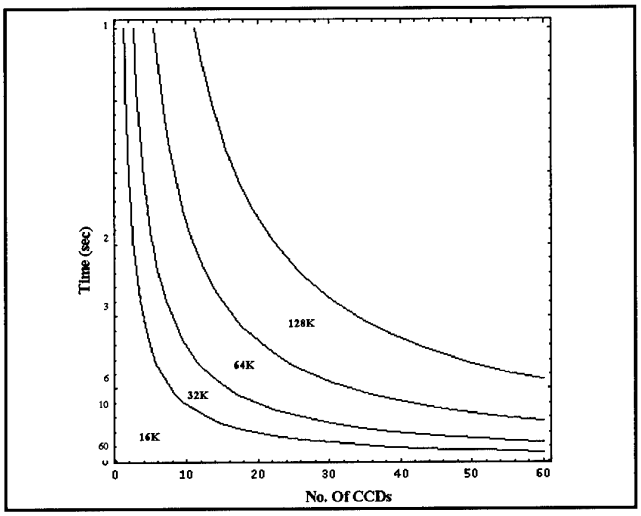

Fig 6.2. Performance of various configurations of the Modular-MPC with regard to the number of CCDs and the acqusition time for objects of size 9 pixels.

Thus, depending on application requirments i.e. CCD read-out rate, maximum object size and the number of objects per sq. deg.), the appropriate size for the system can be chosen. At O.C.A., the 30 cm focal plane of the Schmidt telescope can house up to 16 Kodak 2K x 2K

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

CCDs in a staggered arrangement per column. At sidereal rate, each CCD can be read out completely in 79.8 sec but between 6 -10 times sidereal rate is expected and with object sizes of 9 pixels, a 32k PE system with 32 MBytes memory would suffice. At lower sidereal rates, larger object sizes can be operated on while at faster sidereal rates, smaller objects size can only be handled.

Current research in Aspex aims to develop 16k PEs per board and 16 MBytes of memory on each DSM-SDS board, and DSM-PDR boards stacked on an ISM/DSM mother board. A full SIMD configuration would comprise 4 MPPs, 4 DSM-SDS and 2 DSM-PDR boards on the ISM/DSM board providing a 64K PE system with 64MByes of memory. Boards would be of standard 6U sizes and this configuration would comfortably fit into an industrial workstation e.g. HP9000 748i.

Table 5.1 summarises system board count necessary for the NEO detection system.

Board Type	Number of Boards	Comments
ISM/DSM	1	
MPP	2	16k PEs/board using 4 x 4K PE MCM-Ds
DSM-SDS	2	16 MBytes/board with upto 16 data channels
DSM-PDR	0	

Table 5.1 Modular-MPC configuration for NEO detection

Planned upgrades for next generation ASP modules target factor-of-four increases in the number of PEs through stacking and 3D techniques. With the first technology upgrade, the number of MPP boards would reduce to 1. Advances in high density memory packaging technology should also reduce the number of DSM-SDS boards to 1 and overall board configuration to 1 MPP, 1 DSM-SDS and 1 ISM/DSM boards. Further research into 3D MCM and wafer-scale implementation technology, would ultimately integrate the MPP, the PDR and the SDS into a single package, resulting in a single board system for real-time NEO detection.

Thus, it can be conclusively stated that the Modular-MPC approach to NEO detection problem can provide the necessary performance and cost-effectiveness.

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

References

[AST94] Astromed, *The Role of charge coupled devices in low-light level imaging*, Tech. Guide. 1994

[BIJ92] Bijaoui, A., *Optimisation de la detection*, Tech. Rep. Aug. 1992

[CHA94] Chapman, C., *Live crash from Jupiter*, New Scientist, March 1994

[CRA94] Craine, E.R. *Optically multiplexed two dimensional imaging arrays*, SPIE, Vol. 2198.

[FOR94] Fordham, J.L.A., Bone, D.A., Michel-Murillo, R., *Development of the BIGMIC photon counting detector for astronomical applications*, SPIE, Vol. 2198, 1994

[GEA91] Geary, J.C., Luppino, G.A., Bredthauer, R., Hlivak, R.J., Robinson, L., *A 4096 x 4096 CCD mosaic imager for astronomical applications*, Charge Coupled Devices and solid state optical sensors II, SPIE, Vol 1447, 1991

[GUN87] Gunn, J.E., Emory, E.b., Harris, F.H., Oke, J.B., *The Palomar Observatory CCD camera*, PASP, Jun. 1987, 518-534

[HAN94] Han, W., Clayton, M., Walker, D.D., *Versatile mosaic CCD controller design*, SPIE, Vol 2198, 1994

[JAN87] Janesick, J., Blouke, M., *Sky on a chip: The fabulous CCD*, Sky & Telescope, Sept. 1987

[JOS94a] Joseph R.J., *Near-Eath object detection using the Associative String Processor : Requirements definition*, Tech. Rep. Jun. 1994

[JOS94b] Joseph, R.J., *Notes on the study of convolution techniques on the ASP*, 1994

[KEN92] Kent, S., Munn, J., Petravick, D., *Sloane Digital Sky Survey: Functional specification for survey operation*, Tech. Rep., Nov. 1992

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

[KRI91] Krikelis, A., *Computer vision applications with the Associative String Processor*, J. Par. and Dist. Comp., 1991, Vol. 13,170-184

[LEAC88] Leach, R.W., *Design of a CCD controller optimised for mosaics*, PASP, Oct 1988, 1287-1295

[LEA88] Lea, R.M., *ASP: A cost-effective parallel microcomputer*, IEEE Micro, Oct. 1988, 10-29

[LES94] Lesser, M.P., *Improving CCD quantum efficiency*, SPIE, Vol 2198, 1994

[LUP94] Luppino, G.A., Bredthauer, R.A., Geary, J.C. *Design of an 8192 x 8192 CCD mosaic*, SPIE, Vol.2198, 1994

[WHI94] Whitaker, M., *APE processing node for the WASP system testbed*, Tech. Rep, Nov 1994

[MAS88] Massey, P., *A user's guide to CCD reductions with IRAF*, Tech. Rep., Feb. 1989

[MAU93a] Maury, A. *Designing CCD arrays adapted to NEO searches on existing Schmidt telescopes*, Personal communication, 1994

[MAU93b] Maury, A. *Constuction of a new CCD controller adapted to NEO searches*, Personal communication, 1994

[MOR92] Morrison, D., *The Spaceguard Survey: Report on the NASA International Near-Earth Object Detection Workshop*, Jan 1992

[NIN94] Ninkov, Z., Tang, C., Backer, B., Easton, R.L., *Charge Injection Devices for use in Astronomy*, SPIE, Vol. 2198, 1994

[REI94] Reiss, R., *Array Control Electronics (ACE): ESO's next generation CCD controllers for the Very Large Telescope*, SPIE, Vol. 2198, 1994

[SAV94] Savalle, R., *Conception du logiciel de detection automatique d'asteroides des GODS*, Rapport de stage Inginieur, Oct 1994

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England

[SEK92] Sekiguchi, M., Iwashita, I., *Development of a 2000 x 8144-pixel mosaic CCD camera*, PASP Vol. 104, Spet. 1992, 744-751

[STE87] Stetson, P., *DAOPHOT: A computer program for crowded-field stellar photometry*, PASP, Mar. 1987

[STO94] Sotver, R.J., Brown, W.E., Gilmore, D.K., Wei, M., *Characterisation of a 4K x 2K three-side buttable CCD*, SPIE, Vol 2198, 1994

[VAL82] Valdes, F., *Faint Object Classification and Analysis System*, Tech. Rep., Oct. 1992

[VAL88] Valdes, F., User's guide to the CCDRED package, Tech. Rep., Feb. 1988

[VER91] Verschuur, G.L., *The end of civilisation*, Astronomy, Sept. 1991

[WAL90] Waltham, N.R., van Breda, I.G., Newton, G.M., *A simple transputer-based CCD camera controller*, Instrumentation in Astronomy, SPIE Vol. 1235 1990

[WIN94] Winzenread, R. *Flat thinned scientific CCDs*, SPIE, Vol. 2198, 1994

Copyright Reserved, Aspex Microsystems Ltd., Uxbridge, England