XMM OPTICAL MONITOR

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON   Authors: D.A.Bone, H.E.Huckle

SOFTWARE SETUP OF THE BLUE DETECTOR ELECTRONICS

DOCUMENT:XMM-OM/MSSL/SP/0077.02

Distribution:

XMM-OM Project Office	A Dibbens	Orig
ESA PX	H Eggel	
CSL	P Rochus	
	S Roose	
Los Alamos National Laboratory	C.Ho	✓
UCSB	T.Sasseen	✓
Royal Greenwich Observatory	R Bingham	
University College London	J Fordham	
Mullard Space Science Laboratory	R Card	
	M Carter	
	R Chaudery	
	R Hunt	
	H Huckle	
	H Kawakami	
	T Kennedy	
	D Self	
	P Smith	
	P Thomas	
	M Whillock	
	K Mason	
	A Smith	

Author: Date:

OM Project Office   Date:

Distributed: Date:

CHANGE RECORD

Issue	Date	Comments
1	15-May-1995	EOB2 Description
2	23-April -1999	FM Update - reformatted for MS Word.

Contents

sub-address	port name	port operation	port size
2	Camera_mode	Camera start/stop	1
3	Map_address	Bitmap address	16
4	Map_data	Bitmap data	4
5	H_Threshold	Event detect threshold	8
7	Table_mode	Table access mode	1
8	Table_address	Lookup table address	16
9	Table_data	Lookup table data	6
10	Acq-mode	Acquisition mode	3
		Frame tag enable	1
11	Int_enable	Integration enable	1

Table 1: Blue module writable ports

sub-address	port name	accesses	port size
2	Proc_status	status word	8
3	Map_read	Bitmap data	4
4	Table_read	Lookup table	6

Table 2: Blue Module readable ports

1. Introduction

This document describes the Blue Detector processing electronics from the software point of view, illustrating the procedures needed to set up camera formats and load the various Lookup tables. The features of the EOB-2 electronics are detailed in the document SP56 (Blue Detector Processing Electronics detailed design). This release describes the FM. The Processing Electronics is controlled by the ICU over the MACS bus. A single MACS bus Slave Module handles the traffic to a maximum of 32 ports (sub-address)within the processing electronics. These ports are mostly either read only or write only, and in general, it is not possible to read back a written value. Ports 0 and 1 are used internally to the MACS bus slave module for initialization.

2. MACS bus Sub-address allocation

The write only ports within the Detector Processing electronics are shown in table 1 with the assigned sub-address value. Table 2 shows the read only ports with their sub-address - these overlap the write only ports. Some data values, for convenience within the electronics, end up being read, in some form, from a different sub-address from which it was written.

This is an important note on the error detection by the MACS bus slave module in the processing electronics. The electronic details can be found in the document “Blue Detector Processing electronics detailed design”. For MACS ‘RD’ instructions, the slave module checks for errors in the data word and if one is found, a write strobe will *not* be issued to the port. In this case the ‘RD’ instruction can be reissued as part of the error recovery by the master. However for ‘TI’ instructions, the slave module issues a read strobe to the port in the processing electronics then transmits the data to the master, taking no further part in the proceedings. The master checks for errors in the data word. Of course, if an error is found, it can reissue the same 'TI' instruction and in most cases one will receive the required data. For ports that have the auto-increment feature this will mean the first transmission instigates an increment of the relevant address port and that the error re-read will get the data *from the next* location instead. In this case the address pointer would have to be reset in order to get the original data word.

In the following descriptions of the individual ports and their bit allocations within the 16 bits MACS bus word, bit 0 is the least significant bit and a bit value of 'X' is taken as a don't care on write while being unknown on a read cycle and thus should be ignored.

2.1 Port Camera_mode; Sub-address 2

Camera-mode

X	X	X	X	X	X	X	X	X	X	X	X	X	X	R	S
15															0

Here, 'S' is set to one to start the CCD camera reading out frames with the format loaded into the camera bitmap. The clocking sequence starts with the frame transfer phase. When set, all accesses to the camera bitmap are disabled. Data acquisition can be performed only when this bit is set. Its value is available in the overall status word.

With bit 'S' cleared to zero, the CCD clocks are forced to a standby state and no frames are readout, so no data acquisition is possible. It takes effect immediately. this is the default state when the processing electronics are reset (power up etc.). Only in this state is it possible to access the bitmap RAM to update the camera format, as later described.

When bit 'R' is set to 1, the blue camera head electronics is reset.

2.2 Port Map-address; sub-address 3

Map_address

Bitmap Y address								Bitmap X address							
15							8	7							0

This port holds the 16 bit address of the location within the camera format bitmap RAM that will be accessed next via the Map_data or Map_read ports. It is write only. The least significant 8 bits correspond to the X position within a CCD format whilst the 8 most significant bits correspond to the Y position, or row number. The camera electronics accesses the bitmap during the readout phase with 9 bit counters but with only the most significant 8 bits, corresponding to this port, connected to the RAM (64k) address lines. Hence the bitmap deals with the units of pairs of CCD pixels or pairs of CCD rows. The RAM is not fully utilized for any camera format - the programming is described later.

2.3 Port Map_data; sub-address 4

Map_data

15												3			0
X	X	X	X	X	X	X	X	X	X	X	X	Map data			

This port is used to write data into the bitmap RAM which is 4 bits wide. Data is read back via another port. The location accessed is specified via the map_address port and the least significant 8 bits (X position) of the RAM address auto-increment after each map_data access, thus avoiding the need to re-specify the bitmap address. The data transferred are window ID numbers or CCD row pair action codes.

2.4 Port H_threshold; sub-address 5

H_threshold

15								7							0
X	X	X	X	X	X	X	X	Event Height Threshold							

This port holds the threshold value for event detection. Events are detected if their peak CCD pixel value is greater than this value. Typical values are 30 for normal integrations and 8 for pulse height distribution building. The value loaded into the port should be the two's complement of the threshold required. (e.g. 0xf8 for a threshold of 8).

2.5 Port Table_mode; sub-address 7

Table_mode

15															0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	TE

This port is used to control the access mode to the Lookup table RAMs. With bit 'TE' cleared, which is the default state on power up, the tables are accessed by the processing electronics and normal data acquisition can take place. All other access to the RAMs by the ICU is disabled.

When 'TE' is set to one, access by the ICU to the lookup RAMs is enabled and data acquisition is not possible (it is disabled). Both X and Y tables are accessed in parallel via the Table_data and Table_read ports.

2.6 Port Table_address; sub-address 8

Table_address

Lookup Address (M)	Lookup Address (N)
15 … 8	7 … 0

This port holds the address of the location within the 64K Lookup table RAMs that will be accessed next via the Table_data or Table_read ports. Both X and Y tables have the same address, being accessed in parallel. The tables hold the sub-pixel values to output for each possible event position within a CCD pixel and the contents are in general different for the X and Y coordinates - the programming of the contents is described later. The least significant 8 bits correspond to the N centroid information whilst the most significant correspond to the M information (signed 8 bit). The port is write only.

2.7 Port Table_data sub-address 9

Table_data

X	X	X	X	X	X	X	X	X	Y sub-pixel	X	X sub-pixel
15									6 … 4		2 … 0

This port is used to write data, which are X and Y sub-pixel values, into the X and Y Lookup table RAMs. The location to access is contained within the Table_address port, and the access mode must be enabled via the Table_mode port. After each access, the 16 bit RAM address held in the Table_address port is auto-incremented to point to the next location, thus reducing the need to load address values.

2.8 Port Acq_mode; sub_address 10

Acq_mode

X	X	X	X	X	X	X	X	X	X	X	X	FE	int mode		
15												3			0

This port controls the acquisition mode or the type of format obtained upon integrating. The field 'int mode' determines this as follows;

mode 0. Low resolution (4 sub-pixels per CCD pixel) windowed image data acquisition.

mode 1. Low resolution, full frame image data acquisition.

mode 2. High resolution (8 sub-pixels per CCD pixel) windowed image data acquisition.

mode 3. High resolution, full frame image data acquisition.

mode 4. Engineering measurement of X M-N and Y M-N parameters.

mode 5. Engineering measurement of Y M-N and X M-N parameters.

mode 6. Engineering measurement of event Height and Energy distributions.

mode 7. Engineering measurement of event Energy and Height distributions.

The mode should not be changed while an integration is active to avoid synchronization problems with the data stream in the FIFO buffers.

The bit 'FE' is cleared to zero to *enable* the insertion of frame tag words into the FIFO buffer and thus the data stream. When the first active window in a CCD frame starts a word of all zeros will be inserted in the stream. When set to one *no* frame tag words are inserted. This option is normally disabled for engineering modes and the full frame modes since data words of all zeros can occur here or the tags are not required.

2.9 Port Int_enable; sub-address 11

Int_enable

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	IE
15															0

The bit 'IE' is set to one to enable integrations, where data is allowed to be written into the FIFO buffers. The start is synchronized to the next occurring end of frame transfer phase of the CCD. When cleared, the integration stops on the next frame transfer phase and writes to the FIFO buffers are disabled. the default state is zero after reset (power on).

2.10 Port Proc_status; sub-address 2

Proc_status

X	X	X	X	X	X	X	X	X	int mode	FE	X	TE	ME
15								7					0

This read only port is used to gain the overall status of the processing electronics. The bit ‘ME’ is set to one if the CCD camera clocks are running and cleared if halted and access to the bitmap RAM is possible. Bit ‘TE’ is set to one if the processing electronics have access to the Lookup table RAMs and cleared if ICU access to the tables is enabled. Bit ‘FE’ is cleared to zero if the insertion of frame tag words into the data stream is enabled. The field ‘int mode’ is a copy of the Acquisition mode set for integrations as detailed in the Acq_mode port. Bit ‘IA’ is set if an integration is active and event data is allowed to be written into the FIFO buffers.

2.11 Port Map_read; sub-address 3

This read only port is used to read back the contents of the bitmap RAM once access is enabled via the camera_mode port. After each access the 8 least significant bits of the 16 bit RAM address held in the Map_address port are auto-incremented to point to the next location from which the next read will occur.

Map_read

X	X	X	X	X	X	X	X	X	X	X	X	map data
15												3 0

2.12 Port Table_read; sub-address 4

Table_read

X	X	X	X	X	X	X	X	0	Y sub-pixel	0	X sub-pixel
15									6 4		2 0

This read only port is used to read back the contents of the X and Y centroid Lookup tables once they are enabled for access via the Table_mode port. The address of the next location in the 64K RAMs is held in the Table_address port and this is auto-incremented after each read of the Table_read port.

3. Programming a Camera format

Before the Blue Detector Processing electronics may be used the Camera bitmap RAM must be loaded with information that will cause CCD pixels within the desired windows into the image to be readout. In other words, a clocking sequence is generated for the required camera format. For further details behind the clocking sequence, refer to the document SP56, the electronics description.

In preparing a format CCD pixel coordinates are used. The CCD is 385 pixels in length in the X direction (along a row) and 288 pixels, or rows, in the Y direction. The origin (0,0) is the first row/pixel to be read out of the CCD. The electronics use 9 bit counters to count the rows and pixels but only the 8 most significant bits are connected to the address lines of the bitmap RAM so that a 64K RAM may be used. The least significant 8 bits of the 16 bit address (which can be preset via the Map_address port) are fed from the X pixel counter while the 8 most significant are fed from the Y row counter. The smallest unit that can be addressed is a pair of rows or a pair of pixels. Thus an address of 204 Hex has information for rows 4 and 5, pixels 8 and 9 (shift right one bit the counter value to obtain the address). The size of the CCD implies that the range of the X address is from 0-192 (unsigned 8 bit) and that of the Y address is 0-143 (unsigned 8 bit) so there are unused RAM locations that do not need to be accessed.

For each pairs of rows (even-odd) in the readout phase of the CCD, the camera logic needs to know what to do with the row as a whole, before a horizontal readout starts. Some of the previously unused RAM locations contain this information in the form of ‘row action codes’. For each Y address the X address offset to the code location is constant - ‘action_offset’. The current offset is F6 Hex. Thus the value 2F6 hex holds the action code for both rows 4 and 5. The action code values are;

0 Perform vertical transfer only - no horizontal readout - used for getting rid of unwanted rows.

2 Readout the row but ignore Window IDs, thus dumping unwanted charge buildup.

3 Readout the row taking note of any Window IDs - pixels are used in search for events.

+8 Complete any horizontal readout and skip directly to the start of the frame transfer phase - all following rows in the CCD are dumped into the horizontal register.

During a row readout, the RAM contents are interpreted as Window IDs. The detector area can be divided up into a collection of Windows of varying size. Each Window is assigned a number from 1 to 15 which is the Window ID. All the pixels in a window have the same ID value written into the corresponding locations in the bitmap RAM. A value of 0 is taken as meaning that that pixel pair is not in any window - all the valid locations of a row must be written to. There are restrictions on the placing of windows within the Detector area - only within the bounds of these windows will events be processed. Windows must be a multiple of two pixels in both width and height and cannot overlap in X or Y though they may adjoin. Windows must start at an even number in X and an odd number in Y. This means that row 0 and pixel 384 cannot be used in windows.

Having determined the windows required, there comes the mechanics of loading the bitmap. Fig. 1 shows an example format to work on. In order to access the bitmap, port Camera_mode must be cleared first. It is probably best to deal with the row action codes first - an address is written to the Map_address port, and then an action code to the Map-data port. Out first window starts at CCD row 7 so the format opens with transfer only action codes for the first few rows, accumulating unwanted charge in the horizontal register. Such rubbish must be cleared before a window can start so the action code for rows 4 and 5 is a readout and Dump (Y RAM address 2) where the bitmap contents are ignored for the whole of the line (there is no need to program them with zeros etc). This example illustrates another feature or requirement. For a window to be able to process events cleanly right at the edge of the required window there must be a 1 pixel/row border around the window containing clean data from the CCD. In the X direction there is no problem, provided the window doesn’t start at 0 or end at 385, as every pixel in a row must be read cleanly out of the CCD regardless of whether it is within a window or not. Thus there are no extra programming requirements to ensure borders in X. However, in Y, two extra rows, not in this window, must be readout so that there is data available for the border regions. If these rows are not being readout for another window then they must be readout specially, but the rows will contain no window IDs (cleared to zero). Also, the electronics require that the Window IDs pertaining to a row are actually loaded into the bitmap in the next row up. This gives rise to the 3 boxes shown in the example. This ‘peculiarity’ vitally saves some circuitry in the electronics. Our first window is at row 7 - we have already dumped the charge and we next need to provide for the border regions. The action code for RAM Y address 3 is for a full readout so that CCD rows 6 and 7 will be fully readout. The full X length of the bitmap row needs to be cleared to function correctly. Now CCD rows 8 to 13 hold Window IDs, taking into account the required one row offset in Y for Window ID loading of the bitmap, so RAM locations of Y from 4-6 require action codes of a full horizontal readout. The appropriate locations in X will be loaded with 1 whilst the remainder outside the window are cleared. This gives a window using CCD rows 7-12 with an ID of 1.

The window with ID 4 shows what is done when there are multiple windows on a line. This window uses CCD rows 11-16 so ID 4 will be loaded into the appropriate X locations in rows 12-17 (Bitmap Y value 6 - 8). The border rows at row 10 are being readout for the first window and so no further programming of the bitmap is needed to give the correct borders for the second window. Because all of the rows from 14 to 23 are used for windows or their borders, the row action codes for the bitmap Y locations 7-11 are for a full horizontal readout. The final window is on rows 27-30 so there is a gap between the windows in Y. Row 26 is used for the border. Gaps between windows are filled with vertical transfers only (no horizontal readouts) thus accumulating junk charge in the CCD horizontal register. A Dump row pair must be inserted before the next window can be used. So in this example, CCD rows 24 and 25 are Dump rows to remove unwanted charge - the action code for RAM Y address 12 being for readout and dump. After the last window, ID 3, there are no more rows required up to the end of the CCD at row 286. For all the remaining rows, there should be a row action code of vertical transfer only right up to and including RAM Y address 143. However, this can be cut short by adding the Terminate action code (8) to that of vertical transfer (giving 8). This gives an immediate skip to the frame transfer phase removing the need to fill the remainder of the used RAM with row action codes. This is used in the example at RAM Y address 17. The Terminate action code should not be added to a full horizontal readout, since though the readout will occur, centroiding will be disabled, resulting in an odd height window.

When the row action codes have been loaded, the RAM X locations 0-192 (corresponding to potentially active pixels) must be filled in for those rows having a full horizontal readout, but not those being dumped. Those pixel pairs not in a window should have bitmap locations of zero - the RAM least significant address byte being the pixel X position shifted right one bit. Otherwise Window ID numbers are written. The auto-increment of the RAM X address on accessing the Map_data port can be used to good advantage here.

With the bitmap programmed, a one is written to the Camera_mode port to start the camera clocks running and disable further access to the bitmap.

There are some other considerations in programming the bitmap. Because the electronics works with row pairs, for ease of design only 287 rows of the CCD are read out (the 288th row is thrown away) maximum, and so CCD row 286 should not be used in a window as it results in an odd height window. In long sequences of vertical transfer (more than 50) extra dump row pairs may need to be inserted. If too much charge is dumped into the horizontal register there is a risk of it overflowing into the window rows so the extra dump readouts keep it from getting too full. This would be particularly important if the CCD is running very warm when greater dark charge is generated. It means that the first rows of a window would have a brighter ‘DC’ level.

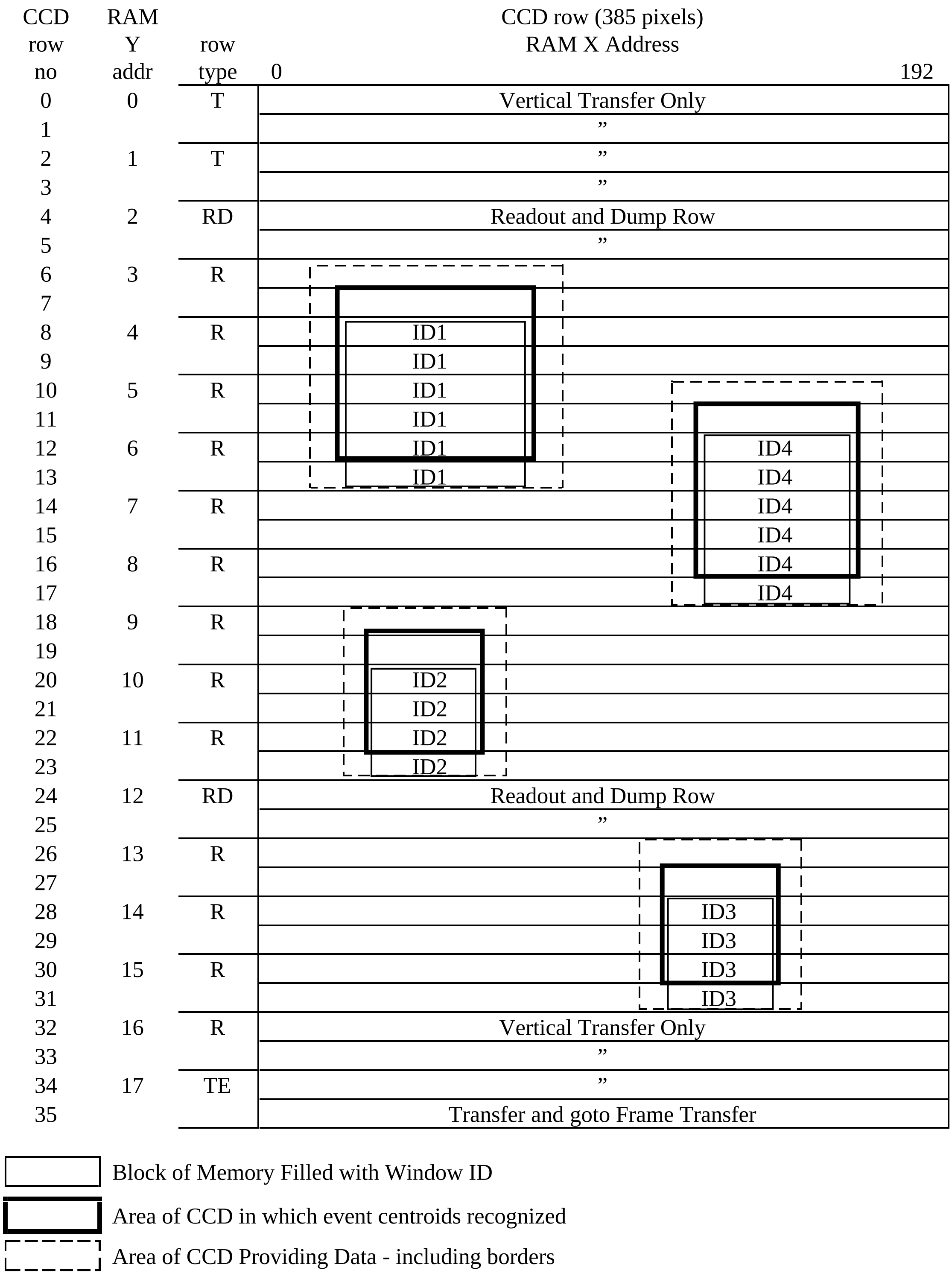

Figure 1: Using the bitmap to give a Camera Format

4. Programming the centroid Lookup tables

Centroiding is the process of locating the position of an event to an accuracy greater than that of a CCD pixel. The processing electronics produce two 8 bit numbers both in X and Y which are input to X and Y Lookup tables. These are **m** and **n**. The division **m/n** is the fractional position within a CCD pixel of the event. The range of positions is divided into 8 bins or sub-pixels. The Lookup table effectively performs the division and binning, outputting the relevant sub-pixel number in the range 0-7. The electronics give a division result in the range - 1.0 to +1.0, where -1.0 is the ‘left’ edge (in X) of the CCD pixel, closest to the origin and 0 is the pixel centre. A sub-pixel value of 0 is given for the left edge.

A routine to prepare the Lookup tables requires two set of boundary values giving the edges of the sub-pixels in X and Y. A single set of 9 numbers could be; -1.0, -0.75, -0.5, ... 0.75, 1.0. Of the two numbers **m** and **n**, **n** is an unsigned 8 bit value that the electronics guarantees is less than 256 while **m** is a signed 8 bit value. Their division gives a centroid value - if this is compared in turn against the entries, starting from the second, in the list of boundary values, until the centroid is less than the list value. A counter following the list position gives the required sub-pixel value directly. We require $list_n \leq centroid < list_{n+1}$. The address of the location in the RAM is given by forming a 16bit word from the bit patterns of **m** (most significant byte) and **n**. The following C code fragment illustrates the process.

```
float x_boundary [9];
float y_boundary [9];
float centroid;
short xsubpix, ysubpix;
int m,n;
unsigned short table_adr, table_data;

for (m = -128; m < 128; m++)
   {
   table_adr = (unsigned short)(m<<8) | 1;
   /* output a start table address */
   macs_out (BPE+TABLE_ADR,table_adr);

   for (n = 1;n < 256;n++)
      {
      centroid = (float)m/(float)n;
      /* find the subpixel values */
      for(xsubpix=0;xsubpix<7 && centroid >= x_boundary[xsubpix+1];xsubpix++) ;

      for (ysubpix=0;ysubpix<7 && centroid >= y_boundary[ysubpix+1];ysubpix++) ;

      table_data = ysubpix<<4 | xsubpix;

      /* output data to lookup tables */
      macs_out(BPE+TABLE_DATA,table_data);
      }
   }
```

This makes use of the auto-increment of the table address on accessing the Table_data port. Of course, better use of this feature could be made on reorganizing this routine. This routine shows that for centroid values less than -1.0, a sub-pixel value of zero is output (the event is actually centered outside the current CCD pixel) and if greater than or equal to 1.0 a sub-pixel value of 7 is given.

In order to load the lookup tables, bit ‘TE’ of port Table_mode must be set so the ICU may have access to the RAMs - no data acquisition can then take place. RAM addresses are set up via the Table_address port and blocks of data for the RAMS (sub-pixel values) are sent via the Table_data port. To give predictable results, even though some address combinations are unlikely, all RAM locations should be written, though **n** zero cannot occur. When finished the Table_mode port should be cleared, blocking out ICU access and allowing data acquisition to take place.

The sub-pixel boundary values are derived from acquisition data sets, usually the engineering mode centroiding data images - some methods are described in the document TC32, ‘Blue Detector Engineering Setup modes’.

5. Output Data formats

The output of the Processing electronics to the DPU is a series of 24 bit words, one per event processed by the electronics. The bit format of the word is determined by the data acquisition mode set in the Acq_mode port. The modes and the bit assignments are shown in Fig.2. There is a set of 4 ‘scientific’ modes and 4 (though really 2 only) engineering modes. The scientific modes provide event position information in the form of the X and Y CCD pixel number, the sub-pixel numbers in X and Y, and the window ID of the window in which it occurred. There is a full frame mode where the window ID is replaced by the most significant bits of the X and Y CCD pixel position counters thus giving 16 tiles covering the full detector area.

The engineering modes provide information for setting up and checking the detector. Modes 4 and 5 capture centroiding information as described in the document TC32, producing a 256 by 256 pseudo-image each in X and Y. From these, sub-pixel boundary lists can be calculated. Due to a convenience feature of the electronics, modes 4 and 5 are equivalent. A single event triggers two words to be sent to the DPU, one with window ID 1 (X) and the second with ID 2 (Y). Modes 6 and 7 give event height and energy (original value divided by 4) in the form of ID images. Again the electronics treats modes 6 and 7 as the same in that two transmissions are made with height given a window ID of 1, energy 2. For the FM, the event energy contains no useful information.

In addition, there is a word of all zeros transmitted at the start of each active frame which can be used for counting frames and other timing. This feature is enabled via the Acq_mode port.

To acquire data in a given mode, the Camera format should be set up as previously described. Windowing is used in all modes but in the full frame mode the window ID does not appear in the data transmissions. However, windows should be defined so that the full detector area is covered so filling the full frame. It is probably best using a single 256 by 256 window of any ID. In the engineering modes Windows of any ID should be defined in order to cover the area of the detector from which events will be gathered so as to provide the engineering information. Again a single large window is usual. The Height threshold should be set via the H_threshold port. In the engineering modes 6 and 7 the Height threshold is set low (8) so as to get a full pulse height distribution, otherwise it is set at the normal centroiding threshold (typ.30). The acquisition mode should be set via the Acq-mode port and the frame tag facility enabled if it is required. The enable bit in the Int_enable port is set to start transmissions to the **DPU** at the start of the next CCD frame. The **DPU** can then acquire an image. When done the enable bit should be cleared to disable further writes to the FIFO buffers, and further transmissions will cease once the FIFOs have been emptied.

In the scientific acquisition modes there are fields of 9 bits only (with sub-pixels) available for the X and Y position of an event. After 2 or 3 sub-pixel bits come 7 or 6 bits from a modified CCD pixel or row counter. The full size of the CCD is 385 by 288 pixels but there is a pre-defined detector collection area of 256 by 256 CCD pixels within this, within which the detector windows are located. The remainder should not be used for windows as they are not properly illuminated with events from the detector image intensifier. This is illustrated in Fig.3. The position of the collection area within the CCD is variable as it depends on the accuracy of the mating of the CCD to the image intensifier and the position of the detector optical axis, but it is nominally central. The position of the origin of this area with respect to the CCD origin is ‘Pix_offset’, ‘Row_offset’ (9,7 in example).

Detector Data Transmission Formats

	Detector Mode	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Science	0	0	YC 6-0							y 2	y 1	XC 6-0							x 2	x 1	ID				P	Low Resolution, Windowed
Science	1	0	0	YC 5-0						y 2	y 1	0	XC 5-0						x 2	x 1	YC 7	YC 6	XC 7	XC 6	P	Low Resolution, Full Frame
Science	2	0	YC 5-0						y 2	y 1	y 0	XC 5-0						x 3	x 2	x 1	ID				P	High Resolution, Windowed
Science	3	0	YC 5-0						y 2	y 1	y 0	XC 5-0						x 3	x 2	x 1	YC 7	YC 6	XC 7	XC 6	P	High Resolution, Full Frame
Eng.	4	0	0	M value								0	N Value								0	0	0	1	P	Engineering, X M/N Data
Eng.	5	0	0	M value								0	N Value								0	0	1	0	P	Engineering, Y M/N Data
Eng.	6	0	0									0	Event Height								0	0	0	1	P	Engineering, Event Height
Eng.	7	0	0									0	0								0	0	1	0	P	Engineering, Event Energy (contains no meaningful data)

P = Parity (Odd)
XC = X CCD Pixel Co-ordinate modulo 64 (high resolution) or modulo 128 (low resolution)
YC = Y CCD Pixel Co-ordinate modulo 64 (high resolution) or modulo 128 (low resolution)
x = X Sub-Pixel Bit
y = Y Sub-Pixel Bit
ID = Window ID

Figure 2: DPU Data Transmission Formats

The modified X and Y counters used in **DPU** transmissions are the result of subtracting these offsets, taking into account propagation delays through the electronics, from the CCD absolute row and pixel counters, so that at the origin of the collection area, the X and Y position fields are zero. The **DPU** collects events in this modified co-ordinate frame but a conversion, using these offsets, to the absolute CCD coordinate frame has to be made when programming the camera format. Event positions are referred to the origin of the collection area and not to the window that they are in, and because of the 9 bit fields, the address ‘wraps around’ so the Window ID is used to keep track of the position.

The default setting is Pix_offset = 70, Row_offset = 16 but is alterable in hardware to suit each detector. It is probably more convenient if the offset in X is keep at an even number and that in Y is made an *odd* number to match the requirements on positioning windows. This will mean that there is no single unusable row or column at the start of the area. It should also mean that the **DPU** need not know the absolute position of the collection area, outputting window coordinates relative to the origin of the collection area with the ICU applying the offset when preparing a format.

To relate to the physical orientation of the detector, the frame of Fig.3 is referred to the CCD - it is the view when looking down into the fibre-optic input of the CCD where the readout port is the bottom left of your field of view (i.e. as if one were looking at a projected image).