Barry Hancock

XMM Optical Monitor

MULLARD SPACE SCIENCE LABORATORY
UNIVERSITY COLLEGE LONDON

Author: D.A. Bone, UCL

Blue Detector Processing Electronics detailed Design

DOCUMENT: XMM-OM/MSSL/SP/0056.03 2-May-95

- NO DIAGNOSTIC EVENT DETECTION.
- NO FRAME SYNC.

Contents

List of Tables

List of Figures

1 General

A prototype design for the Blue Module electronics has been completed to the stage of schematic capture with Viewlogic and is ready for PCB layout and construction. This design, detailed in this document, should not be considered as finalised.

The scope of the design includes the CCD camera controller (to operate the detector head where the CCD clocks are generated), the processing or centroiding electronics to calculate event positions from the stream of digitised data from the detector head, and the necessary control. It also includes the **MACSbus** interface module (a recent addition) and a 'Mother board' to support the cards of the design. It does not include the EHT voltage control for the EHT supply, nor the Monitoring module (temperatures, voltages etc).

The design uses components taken from the GEC-Plessy SOS Radiation Hard July '94 Handbook, only. The relatively low speed of the SOS logic (25ns max delays, 18ns setup times min typically) imply a large degree of pipelining of operations in the design, which is fully synchronous with the CCD pixel clock rate of 10MHz. Worst case propagation delays (for −55 to +125C) were used.

On signal nomenclature, active low signals are indicated by the suffix '*' — on circuit diagrams these are shown by 'barred' names. Within timing diagrams, they are indicated by the 'tilde' prefix.

2 Summary of features

The following is a list of the features of this design — they are described in more detail in the descriptions of the electronics blocks.

- Control of the Detector head clock sequencers to operate the CCD in a frame transfer mode at 10MHz horizontal rate and verticals at 1.67MHz (variations in the rate at the detector head can be coped with).
- Definition of areas (windows) of the CCD under software control which are used to acquire data. Windows are assigned a Window ID number from 1–15.
- Full frame readout of the CCD where one large window is defined.
- Windows may be placed anywhere within the full area of the CCD (385 pixels in X by 288 rows in Y), may adjoin but not overlap. Other restrictions are enforced by software only.
- Windows must be a multiple of 2 pixels in width (X), and a multiple of 2 rows in height (Y). The origin of a window must be an *even* number in X and an *odd* number in rows (Y).
- Window information is held in a bitmap RAM. To load, an address has to be latched and a data word sent. The least significant 8 bits (corresponding to X CCD position) auto-increment on write.
- To change the camera format, the CCD clocks are halted with a local 'reset', and the camera is restarted at the start of frame transfer upon command, after the format is updated.
- Continuous readout of horizontal register during frame transfer (TBD). Gives faster dump of unwanted charge.
- Black level subtraction from each row of CCD data, using the average of the 4 reference pixels at the start of each row.
- Provision of a 'test port' with 26way pinout equivalent to EOB-1 camera head output which can be connected to the EOB-1 frame store and Real Time display.
- Events are recognised by $A \leq B > C$ in both X and Y, and the event peak *height* (B) being greater than a threshold, and the event peak pixel being within a window, at the same time. (This gives a signal, 'ECD', used to enable other circuit blocks).
- The detection threshold is software programmable. (It has a secondary use).

- Windowed integrated data will have no borders of invalid pixels (with correct software — implies extra two rows and two columns outside of the window, will be readout of the CCD).
- Readout sequence of CCD can be terminated early, before all rows are transferred, going straight to the frame transfer phase.
- Event energy is determined to 10 bits plus overflow, but only 8 MSB are used/logged (EOB-1 had no overflow bit).
- Double event counting — flag bit set in transmissions to the **DPU** if an event exceeds (using overflow bit) a software programmable 8 bit value. A control bit enables/disables the feature.
- Event centroiding uses the triangle/parabola algorithm using the pixel data values (not summed in rows or columns) of a 3 by 3 crosshair centred on the event peak.
- Calculates centroiding information (**m**–**n** values) to 8 bits — if during the process the 8 bit range of either is exceeded, both are auto-ranged (divided by two) to keep within the limits of the values supplied to the lookup tables.
- Centroid lookup tables of 64K by 3 translate the **m**/**n** values to sub-pixel values for the events, according to the programmable contents.
- Lookup tables are accessed in parallel during the programming operation so that X and Y take 3 bits each of the word transferred to the tables.
- Lookup table accessed by loading an address latch, then reading or writing data from/to that location. The 16 bit address auto-increments on data access.
- Windows to be limited to within a 256 by 256 area on the CCD by software. This area is selectable by hardware links and applies to the following;
- Final event address is adjusted to have its co-ordinate origin at that of the 256 square defined area. Thus event X and Y addresses equal zero at this point.
- X and Y addresses increase for pixels that take longer to shift out of the CCD (8 bit counters).
- Data acquisition mode is software programmable into two groups — normal integrations and engineering mode. A multiplexer selects the information to be transmitted to the **DPU**.
- Normal integration mode offers high or low resolution (8 or 4 sub-pixels per CCD pixel), windowed or full frame modes, so that up to the full 256 square area can be utilised to give a 2048 square image. Windowed mode includes the Window ID bits in the **DPU** data word while full frame mode uses the 2 MSB of the X and Y 8 bit CCD pixel counters.
- Engineering modes: Event peak height, energy, and X and Y event **m**–**n** information are selected for incorporation into the **DPU** data word.
- 'Pair logic' ensures that event height then energy are transmitted sequentially, or X then Y **m**–**n** information, in engineering mode, to avoid a need for two integrations.
- **DPU** data words are buffered in a 512 word deep FIFO — peak writing rate is 5MHz.
- The FIFO is reset at the start of the first valid row of a CCD frame to reduce potential FIFO problems.
- A frame tag word consisting of all zeros is loaded into the FIFO at the start of the first valid row of a frame. The feature can be disabled under software control.
- Data transmission to the **DPU** is serial with 23 bits plus parity. Maximum rate is 24 serial clock periods (max 5.1MHz) plus two periods interword gap minimum.
- Control via **MACS**bus — occupies 8–10 **MACS** sub-addresses for write operations.
- Readable data over the **MACS**bus includes lookup tables and camera bitmap contents, and a 16 (8 bits defined) bit status port (state of enables, modes etc), only.
- Supports **MACS** bus communications functions 'TI' and 'RD' only.

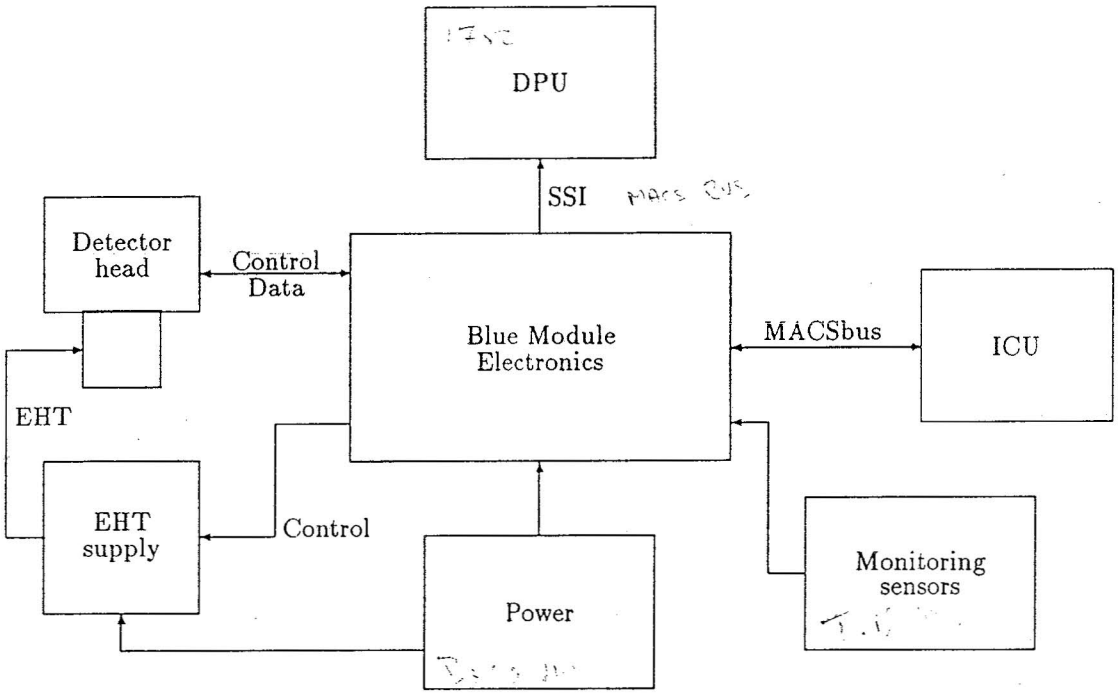

Figure 1: Relationship of the Blue Module Electronics

3 Overview

The relationship of the Blue Module electronics to the other component blocks connected with the blue detector is shown in Fig. 1. The most important block that concerns the Blue Module is of course the detector head containing the CCD and image intensifier. The electronics here that have to be communicated with, are the clock sequence generators (horizontal, image and storage area clocks) and an ADC which digitises the CCD pixel data. This unit is also the source of the 10MHz system clock.

The Blue Module electronics are shown in more detail in the block diagram of Fig. 2. This shows, apart from power, 5 groups of connections to outside modules: Control and data to the detector head, a **MACSbus** interface, a fast serial link to the **DPU**, connections to monitoring sensors, and a link to control the EHT supply for the image intensifier. The operation of the blocks are described in detail in the following sections, apart from the full detail of the monitoring block and the EHT control which are outside the scope of this design. All blocks are driven by the 10MHz clock and are fully synchronous, changes occurring on the rising edge of the clock, with no timing interval less than 100ns.

The Blue Processing electronics are physically laid out on 6 circuit cards, plus a Mother Board. This gives the maximum board area for the volume available. The Mother board supports the 6 cards, interconnecting them — a bus structure cannot be used to connect the cards because of the large number of interconnects that would be required, so the circuit cards occupy fixed slots. Even so, 98 way Hypertac backplane connectors are needed on all cards, with two cards needing a extra 29 way connector to provide the necessary I/O. Some of the connections are taken up by a limited bus structure, with 'discrete' wiring for the remainder. The Mother board also supports connectors to the other modules of the Blue Detectors and the OM wiring Harnesses.

The 6 circuit cards are of the same size except for one which is smaller and designated 'Lookup'. These 6 cards are known as 'Lookup', 'Selector', 'Process', 'Darray', 'Camera' and 'Monitor'. The 'Monitor' board holds all the functions outside the scope of this design. The boards hold the following circuit blocks, chosen to fill the board areas as evenly as possible, avoiding splitting functions between cards and minimising card interconnects:

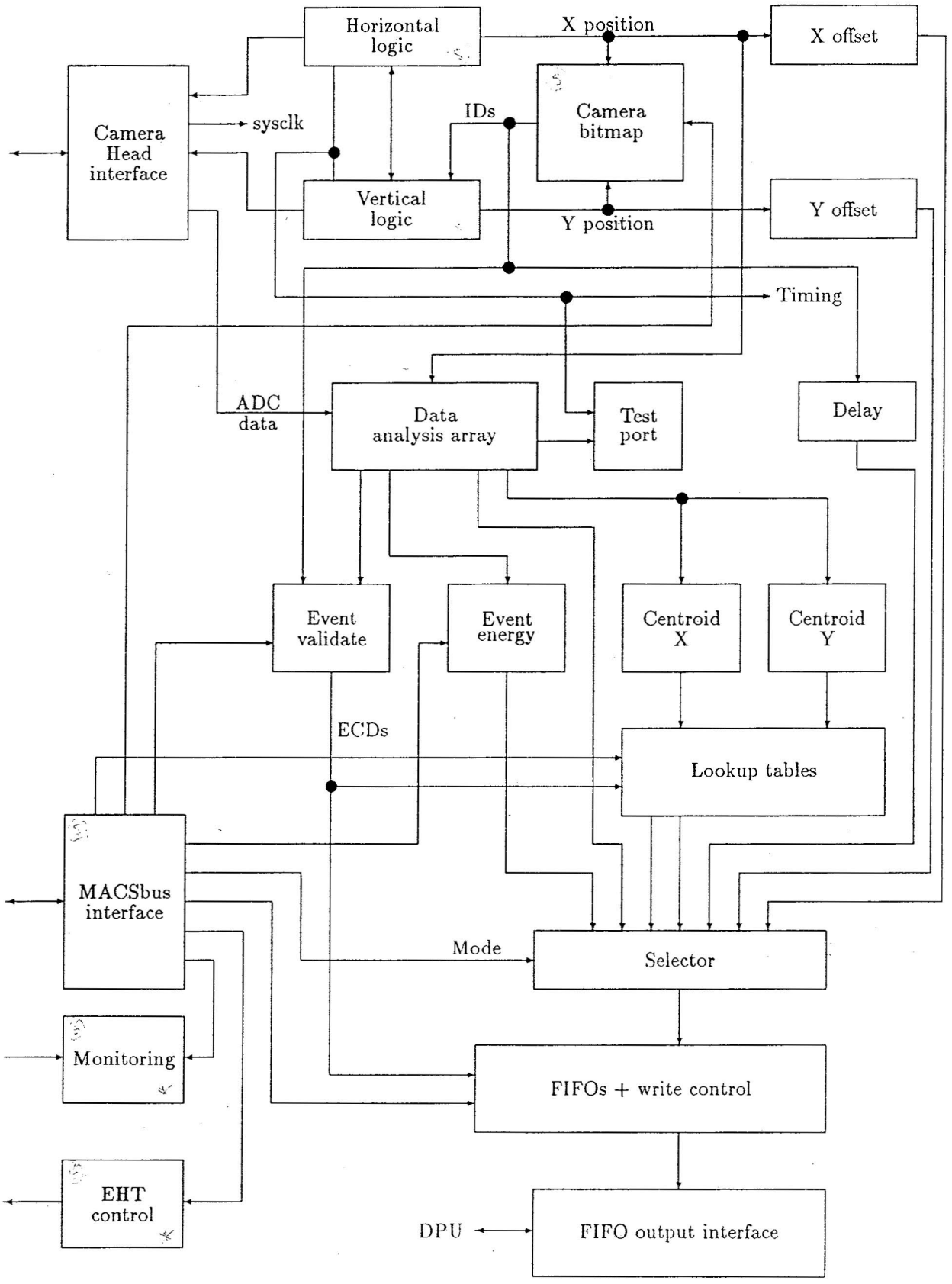

Figure 2: Blue Module electronics block diagram

Signal	Description	I/O to Blue Module
MCLK	10MHz master clock	input
HRUN*	Horizontal clocks enable	output
SRUN*	Storage area clocks enable	output
IRUN*	Image area clocks enable	output
CRES*	Camera reset	output
VD0	Digitised CCD data – LSB	input
VD1	CCD data	input
VD2	CCD data	input
VD3	CCD data	input
VD4	CCD data	input
VD5	CCD data	input
VD6	CCD data	input
VD7	CCD data	input
VD8	Digitised CCD data – MSB	input

Table 1: Detector head interface signals

Camera: MACSbus interface and decoding, Camera Horizontal logic, vertical logic and the Camera bitmap.

Darray: Black level subtraction, Data Analysis array, Test port and the Event Validate block.

Process: Event energy and Double Count logic, X centroid logic and the Y centroid logic.

Lookup: X and Y centroiding Lookup tables.

Selector: X and Y Counter Offsets, the FIFO Data selector/multiplexer, the FIFO write logic and FIFO buffer, and the FIFO read and DPU link logic.

The operation of the circuit modules will be described on a card by card basis. There is a natural progression through the system which would, in general, be followed whilst testing the electronics. The MACSbus unit provides overall control and allows functions to be setup and read back. The Camera logic sequences the detector head (as long as a 10MHz clock is available, the detector head electronics need not be present) and provides timing signals for the other cards. Then the digitised data, either from the CCD or a test pattern generator, flows first through the black level subtraction, then into the data analysis array to produce 3 parallel streams of data to which, in parallel, are attached the Event validate logic, the Event energy calculator and the X/Y centroid information calculators. The centroid information is presented to the Lookup tables, which must be loaded, to produce Event Centroid positions. The various streams of data are presented to the FIFO Data selector and written into the FIFO buffer. Finally the DPU link logic transmits the results from the FIFO buffer.

4 Detector head interface

This small block handles the communication with the portion of the detector head electronics that deals with the CCD, and essentially consists of buffers. The signals include control lines produced within the CCD camera logic blocks of the Blue Module, as well as digitised CCD data, and are listed in table 1. All control lines are active low and change state on the rising edge of MCLK. The actions instigated in the detector head are as follows:

HRUN* At the inactive level the CCD horizontal clocks will be in the standby state. From the transition to the active level, the horizontal clock sequencer should start on the next edge of MCLK and clock out CCD pixels at 10MHz. On the next edge of MCLK after the inactive transition, the sequencer shall halt in the standby state.

SRUN* At the inactive level the CCD storage area vertical clocks will be in the standby state. From the transition to the active level, the storage clock sequencer should start on the next edge of MCLK and thereafter transfer CCD rows one every 6 MCLK periods (to give proposed 1.67MHz rate). On the next edge of MCLK after the inactive transition, the sequencer should halt in the standby state. The length of the active pulse will be a multiple of 6 MCLK periods.

IRUN* At the inactive level the CCD image area vertical clocks will be in the standby state so that the CCD can gather charge. On the transition to the active level, the timing is as for SRUN*. IRUN* will not be asserted without SRUN*.

CRES* This optional signal indicates that the camera is in a reset state, such as at power on reset.

There are some important timing constraints that must be met. These are measured with respect to the rising edge of MCLK on the interface connector of the Blue Module electronics. As seen on this connector, the control signals change state 11–67ns max from the clock edge. The video data, VD0–7, should be setup at the connector 13ns min before the edge and held to 3ns min after. The Video data is clocked into a latch so as to keep consistent signal timing with respect to the local system clock in the Blue Module electronics — this is physically located on the 'Darray' card. The Control signal drivers are located on the 'Camera' card.

The 10MHz clock from the detector head is distributed on the Mother board. Because of the fairly critical timing on the circuit cards it is important to minimise clock skew between the system clock nets of each circuit card or there will be difficulty in transferring data from card to card. There is a high input loading factor of the clock nets on the cards, so this is a critical area. The master clock is fed to several buffers which are located in the same package to reduce delay differences, with each buffer driving one card from the mother board. Any problems with the digitised CCD data setup time gives a jitter of a pixel in the start position of a data row which appears as horizontal stripes in an integrated image.

5 MACSbus and Camera control: Camera card.

5.1 MACSbus interface

This interface block is considered divided in two;

- A MACSbus slave module which sits on the bus and understands the bus protocol. On the slave side (communicating with the Blue Module electronics), it provides 5 address lines (32 MACS sub-addresses), a bi-directional 16 bit data port and simple read and write strobes. This is a recent addition to the scope of the design.
- A decoding section which provides strobe signals (or clocks) for the various ports within the Blue module electronics that require software access.

The MACSbus slave module requires a clock signal — it is conveniently provided by division of the 10MHz system clock to give 5MHz and 2.5MHz frequencies — the latter being a option available if there are difficulties with the MACS bus operating at a clock frequency of 500kHz, when 250kHz can be used instead.

5.1.1 MACS bus slave module

For full details of the MACS bus protocol and applications refer to the 'MACS Handbook' (RD00xx).

Though a Hybrid device is in theory 'available', it is bulky and so the interface to the MACS buslines is made with discrete devices with a logic block to provide clock sync separation and channel selection. This then connects to a MACS controller integrated circuit which handles the actual bus protocol and drives the decoding logic block (described next section).

There are two channels to the MACS bus, A and B, which use identical transceiver circuits. The MACS clock differential lines (eg. MCLKA0, MCLKA1) and data lines (eg. MDATAA0, MDATAA1) are received

by high speed comparator circuits, type LM119, and converted to single CMOS level signals. These are always active. There is no requirement to drive the clock lines but the MACS data lines are driven by discrete transistor buffers using saturated NPN/PNP switching transistors. With the transistors being independently driven by complimentary data out lines (DA+, DA-), it is possible to turn both off when the channel is not active, presenting a high impedance load to that channel of the MACS bus. There are jumper links to select termination resistors for the prototype design.

The received signals are passed to a 2:1 multiplexer to select channel A or B of the MACS bus, depending on which is active. The selected MACS clock, 'SMCLK', is fed into a chain of 2 D latches which sample the signal at a 5MHz frequency (or jumper selected to 2.5MHz if the transmission frequency is only 250kHz). The gates on their outputs allow edge detection, positive and negative, of the bus clock, giving single 200/400ns pulses on a transition. These combine to give the signal 'TRANS*'. Timing for the MACS clock detection is shown in Fig. 3.

To determine the activity of the MACS clock, an 8 bit counter is used. It is cleared each time the bus clock makes a transition and thereafter counts up in 200/400ns intervals. Should the bus clock be quiescent for 51μs, the counter times out (signal TIMEOUT). There is a requirement for MACS bus slaves to monitor the bus clock — if one channel's clock goes inactive, then after a time the Data channel will be deemed inactive and turned off. Then both MACS channels are examined for an active clock. If the clock exists for a time 'Tact', then the clock is considered active and the corresponding data channel is made active.

Timeout causes a JK latch to set so that 'ACTIVE*' goes high, disabling the Data channel (via gates) and turning off the clock to the MACS Controller device. It also toggles a second latch which provides a channel select signal 'CHAN-A', connected to the multiplexer. Thus Timeout disables all channels, switches to the other MACS channel (B in the example), looks for a clock, then if not found, since the counter starts from zero at the switch, a second Timeout forces a switch back to the first channel. The process stops when the first channel to become alive is intercepted. The example shows channel A becoming active in the 3rd period. The gated clock is 'MCLKV' — the controller device performs further validation of the clock waveform structure. Power on Reset gives channel A the first chance. The TIMEOUT signal, delayed by 2 sample clock periods, is used to gate the clock edge detection to prevent the switch over to a differing level on the clock at Timeout from being considered as a clock edge, and thus being considered for activity. When inactive, gating powers off all the data bus driver transistors. The 'SYNC' signal is held high when no clock is present.

The MACS clock is a continuous clock at 500kHz (nominal) but with a missing clock period, when the clock is low, every 25th period. This has to be detected and a 'SYNC' pulse provided for the MACS Controller device — this active high pulse enclosing the next clock rising edge after the missing period. This signal is also high when the bus clock is inactive. Timing for the Sync separation is shown in Fig. 4. If the Timeout counter counts up to 10 (2μs), giving 'MISSCLK', then the clock is either going inactive or is missing a cycle, so a JK flip-flop is cleared, setting 'SYNC'. The latch is only set, clearing 'SYNC', on a rising edge of the bus clock, so generating a pulse of suitable timing as long as the clock is running normally. The timing is such that a 10% variation of the clock period can be accommodated.

To determine if a channel's bus clock is active, the generated SYNC pulses are counted. Falling edges of SYNC (gating the clock transition with SYNC active) are saved in a latch, giving the signal 'SYNCOLD*'. On the second edge (of SYNC2* in Fig. 3), the clock is considered active and the ACTIVE* signal is set low. A Timeout would cause the latched result to be cleared and the search to restart. This gives the bus clock time to stabilise and ensures that there are some features of a MACS bus clock present before it is considered by the bus controller device.

The bus controller device, Type FU-02, receives these signals and provides the user bus. The equipment address, determining which section of the address map is decoded, is set by jumper links on the prototype. The 'HOLD' signal indicates there is to be a transaction on the user bus and the RD* and WR* are used with some gating to provide read and write strobes. The read strobe also switches the direction of the bi-directional 16 bit data bus buffers (sheet Camera 14). All strobes are gated with INS/DATA to select a data transaction, and the write strobe is also gated with 'ERC' so that it is only issued if the transmission completed successfully. An unsuccesful operation won't cause a value change in a port. If a read operation fails then it must be repeated. If, however, an auto-increment port is accessed then the address pointer must be re-written in order to obtain the data value that would otherwise be missed.

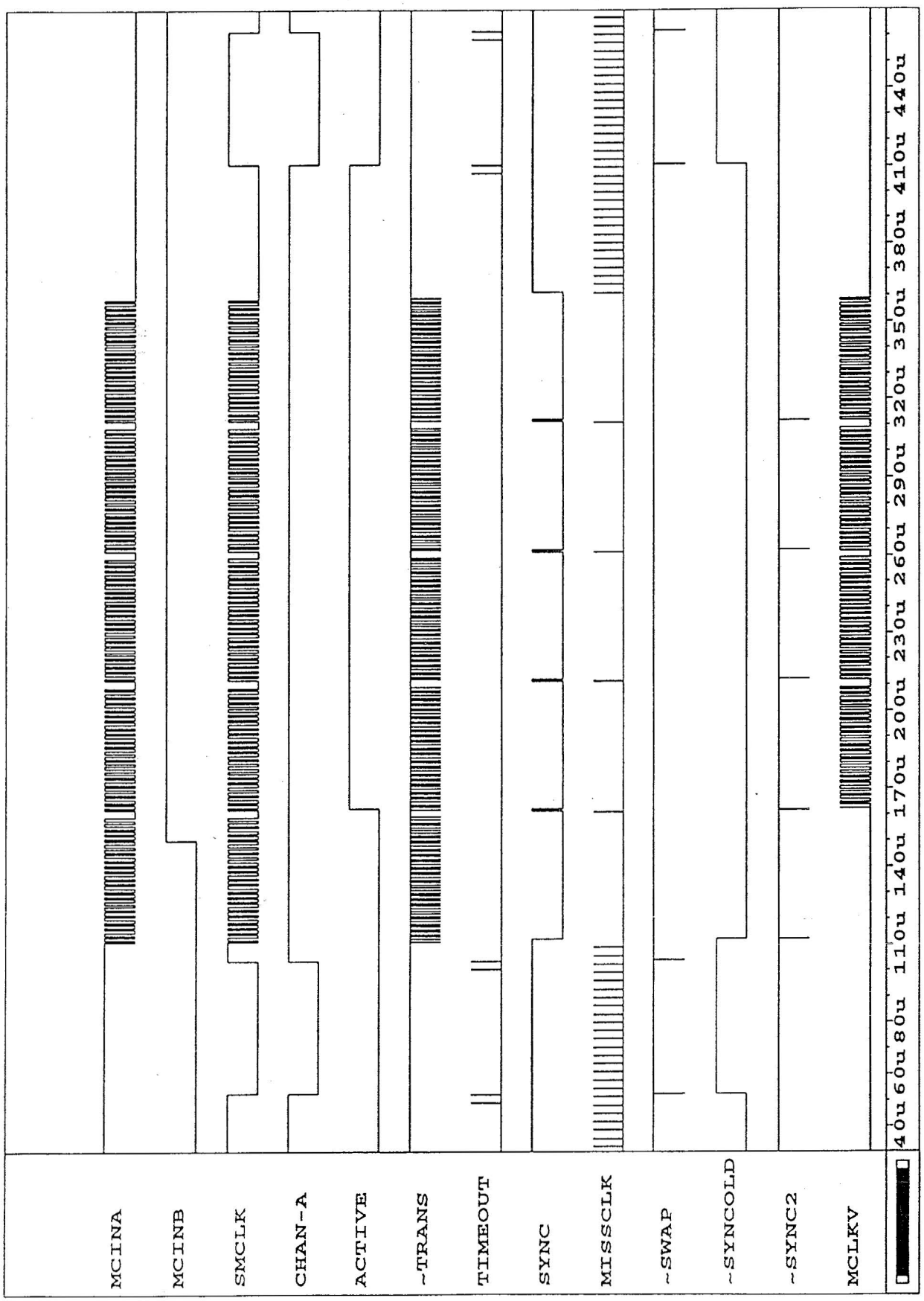

Figure 3: MACS Clock Detection.

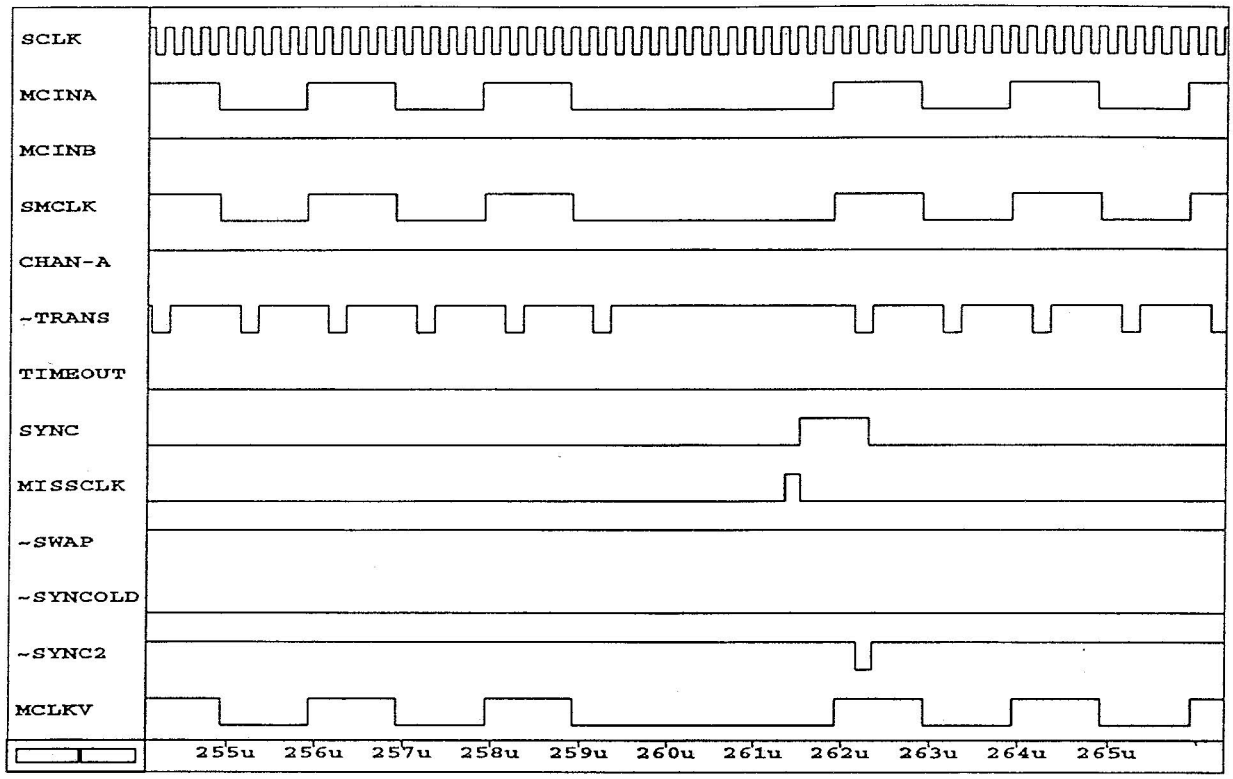

Figure 4: MACS Sync clock separation.

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

Table 2: Blue Module readable ports

Control software should take note of this. The sub-address bus is latched in a transparent latch by RD* (it is set transparent by HOLD low) to ensure the address is stable past the end of the read strobe as the controller device does not guarantee this.

5.1.2 MACS bus decoding

The address decoding is done in two portions; the read strobe enables up to two off, 3 to 8 line decoders (eg. 54HSC138) without further gating. Thus a potential 16 strobe lines are available which can enable various information onto the 16 bit data bus. Otherwise, the write strobe enables a different set of decoders in a similar fashion giving a set of clock/strobes[1] causing a specified port to latch as much of the 16 bit data bus that it requires. There are only 3 ports in this design that give readable information, as shown in table 2. The status word provides quick information on which features in the module are active etc. The available data is shown below:

Camera running	1 bit
Lookup table access mode	1 bit
Double Count enabled	1 bit
Frame tag enabled	1 bit
Acquisition mode selected	3 bits

[1] The write strobes were known as 'OC's in the MIC ground based program

sub-address	port operation	port size
2	Camera start/stop	1
3	Bitmap address	16
4	Bitmap data	4
5	Event detect threshold	8
6	Double event threshold	8
	threshold enable	1
7	Table access mode	1
8	Lookup table address	16
9	Lookup table data	6
10	Acquisition mode	3
	Frame tag enable	1
11	Integration enable	1

Table 3: Blue module writable ports

Integration in progress 1 bit

There are 10 writable locations covered by this design — these are shown in table 3. The unused decoded strobe lines are available for accessing ports in the Monitoring block or the EHT control block. These are 'RD P5–7*' (read) and 'OC P12–15*' (write).

Timing is arranged such that data should be latched on the rising edge of a Write strobe, and should be valid on the bus MBD[15:0] 1630ns before the rising edge of a Read strobe, remaining until completion of the strobe. The write strobe is present for half a **MACS** clock period while the read strobe is present for one period. Full details of the bit order and how to set up the ports in the Processing Electronics are in the document 'Software Setup of the Blue Detector Electronics'. This circuit is on Camera 3. A Power On Reset generator is also shown here — this drives all of the processing electronics.

5.2 Camera Horizontal logic

The purpose of this section is to run the CCD horizontal clocks, counting them and stopping when a whole row has been transferred. It also determines which pixels can contain active data and those that are for black level reference. This circuit is on Camera 6.

The pixel sequence for a horizontal readout is shown in Fig. 5 — this is that for an EEV 385 by 288 CCD. This shows that a complete readout sequence consists of 407 pixels of which only 385 are able to receive valid photo-generated charge. Of the rest 4 elements represent the dark current for each row of the CCD image and storage areas and can be used as black level reference pixels ('r' in figure), the '*' are transition elements while the rest receive no direct charge. Refer to the EEV data sheet for more details. All must be read out or charge will be left to corrupt succeeding rows of data.

Fig. 5 also shows some timing signals — IBLK* which indicates which are the black level reference pixels and LSYNC* which are the valid pixels within the line available for use within data windows. Non zero window ID's will indicate valid window data in this period — outside this, LSYNC* forces window ID's to zero. Window ID's are described in the camera bitmap block section.

The horizontal readout sequence is configured to start whenever the CCD vertical clocks are halted, producing an HSTART pulse, thus driving HRUN* low and enabling (by removing the LOAD* input) a 12 bit counter which thus counts CCD pixels. The timing for the start of a horizontal readout is shown in Fig. 6. The rising edge of SRUN* is delayed so that HRUN* is active two clock periods later, giving time for the vertical clocks to settle (see also Camera 7).

When the counter reaches 407 pixels in all, detected by an 8 bit comparator producing 'LINEND*', HRUN* is deasserted on the next clock edge and the readout is complete. This signal is two clock periods long as the comparators ignore the LSB of the counter so 8 bit devices can be used — their enable input

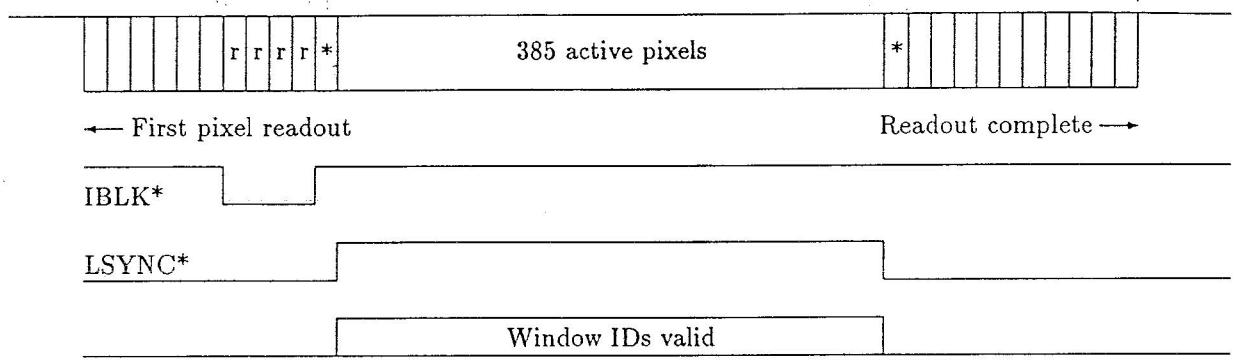

Figure 5: CCD Horizontal Readout

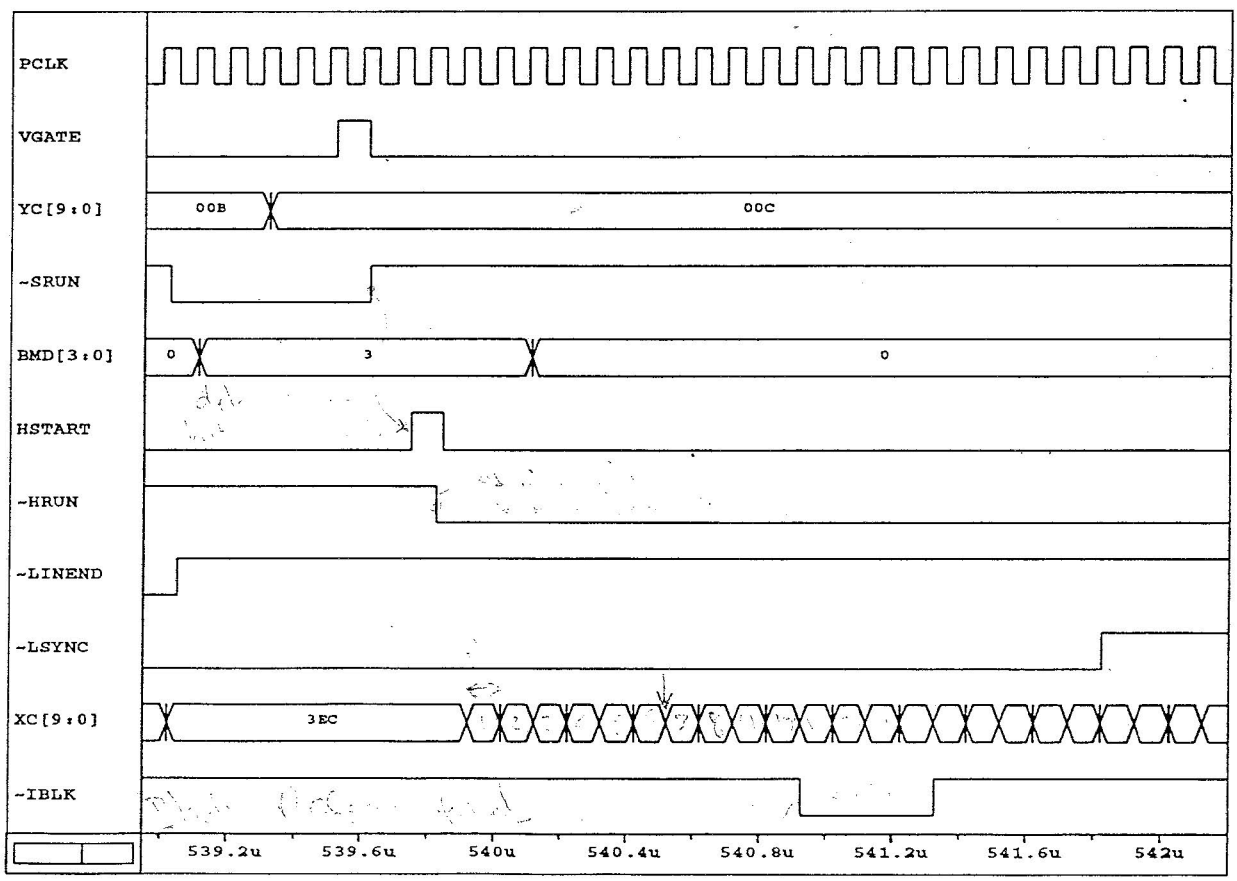

Figure 6: Timing for start of Horizontal Readout.

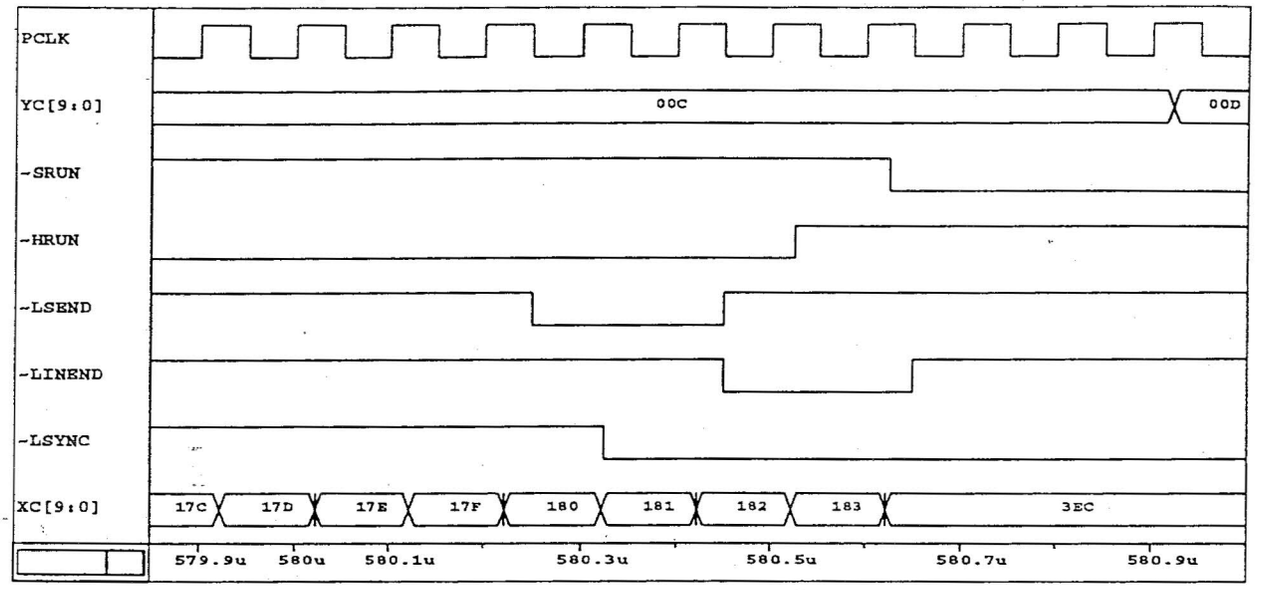

Figure 7: Timing for the end of Horizontal Readout.

is driven only when the counter is positive, giving valid numbers. The end of line timing is on Fig. 7. SRUN* is asserted a clock cycle later, restarting the vertical clocks after the horizontal clocks have settled. The counter is returns to its parallel load value while HRUN* is inactive. The counter load input is gated to allow the bitmap auto-increment feature to function — with 'LRESET*' asserted (when the camera logic is halted and the bitmap is accessible) the enforced load is disabled and control of the counter (load and count enable inputs) is turned over to the bitmap load circuitry.

The counter is also used as control input to some circuit blocks and act as a source in finding the position of an event in CCD pixels. For convenience (and simplifying the circuitry), the counter is pre-loaded with a negative number 'Z' (current value -20) set by hard wiring the counter load inputs using the pull-up/down resistors on AX[1:8]. This offset is set such that on the clock period that valid pixel zero (the first real pixel out of the CCD) is available on the output of the data analysis array, LSYNC* goes high and the counter outputs zero as an address to the bitmap. This implies that location zero holds the window ID for the first active CCD pixel. 'Z' takes into account the propagation delay from CCD clock to CCD output, through the ADC, the detector head interface, the black level subtraction and the data analysis array (ie. $Z < -11$). LSYNC* is brought low after the counter reaches 384, detected by an 8 bit comparator providing the signal 'LSEND*'. LSYNC* and IBLK* are held low/high respectively by a signal 'DUMP*' which indicates the row holds rubbish charge to be dumped, and is generated by the vertical logic with the aid of the bitmap. This prevents any events from being falsely recognised.

A 16 bit shift register on the HRUN* signal is used to generate the IBLK* signal a fixed time after the horizontal clocks start as in Fig. 6 — it is tapped at a point when the black level reference pixels actually arrive at the black level subtraction block. The internal gating within the second half of the register, with the aid of two outputs of the first, creates a pulse of 4 clock periods.

The window ID's are sent to a delay block for use later in the logic chain with the second portion on the Selector card. The ID's are also ORed together to produce a single 'window active' signal (WIN ACT*) to be used in the event validate section. A counter/latch with its load input fed from XC0, the X counter LSB, delays this by 2 clock periods to suit the validation — also allowing it to be forced high when LSYNC* is active low on the clear input, preventing extraneous windows. A full horizontal readout illustrating the use of windowing is shown in Fig. 8 (ID[0:3] are produced on the Selector card).

5.3 Camera vertical logic

This block performs the overall control of the CCD readout sequence, with the aid of the bitmap block. The majority is shown on Camera 7. It contains a counter to count vertical clock transfers and makes

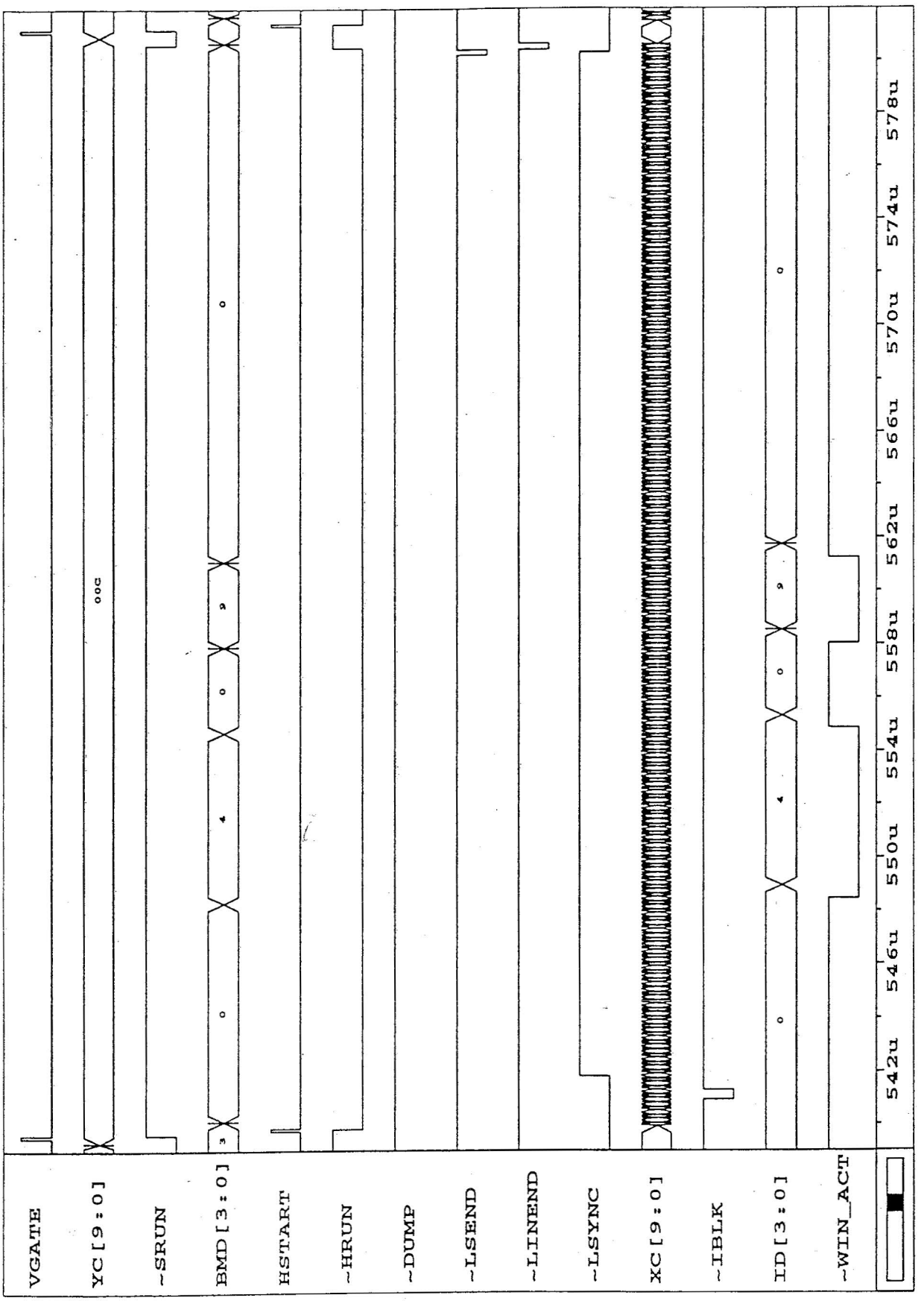

Figure 8: Full Horizontal Readout timing.

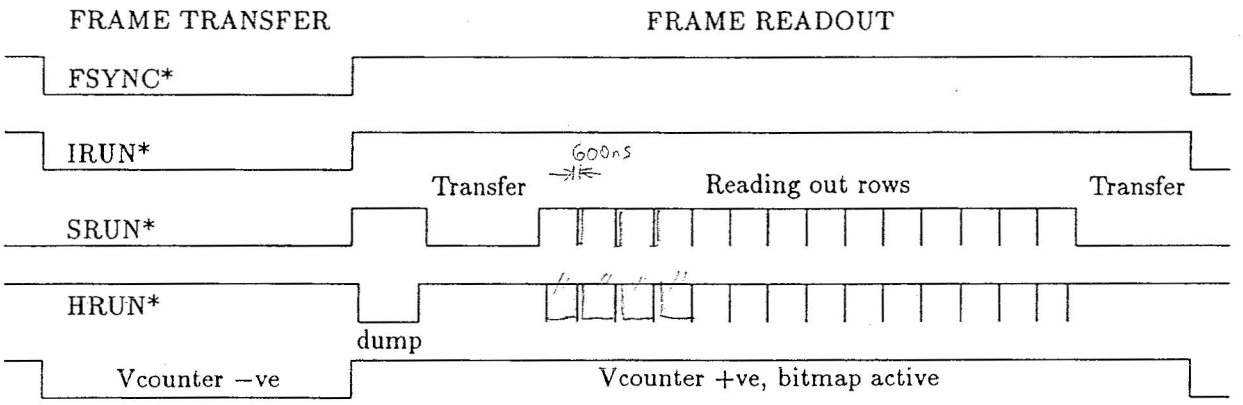

Figure 9: A Frame readout sequence

decisions on a row by row basis. The sequence of events for reading out one CCD frame is illustrated in Fig. 9.

The sequence illustrated shows 14 rows being readout of the CCD — not all of them will be used for collecting events, as discussed in the bitmap section. It shows there are two phases to reading out one frame of CCD data; Frame transfer and the storage area readout phase. Let us start with Frame transfer. This happens to be the point at which the CCD starts after a reset etc. If neither HRUN* or SRUN* is active then SRUN* is forced active (this provides a convenient method of restarting after reset). As the Vertical clock counter is negative at this point, IRUN* will simultaneously be asserted so that the CCD image and storage area clocks run in phase transferring an integrated image into the storage area of the CCD. The vertical clock counter counts out, in the case of an EEV CCD, 290 clock phases and when it reaches −1, both IRUN* and SRUN* are deasserted. An image is now at the bottom of the storage area and the CCD image area can now 'integrate' another frame of data. The storage area readout phase now starts.

Charge that was in the storage area (mostly dark current) has now piled up in the horizontal register, possibly overflowing it. This has to be removed before real data can be loaded in, so the end of frame transfer (actually the action of SRUN* deasserting) triggers a single 'dump' readout of the horizontal register (no LSYNC* is generated so window ID's are not active). On dump readouts, no black level reference signal is generated as the relevant pixels are likely to be contaminated. An option could be considered — that is running the horizontal clocks continuously during frame transfer to help clear the charge and allow fewer dump row complete readouts afterwards. The horizontal register will be cleaned more quickly and there will not be a large 'transient' at the CCD output in the frame transfer phase, but at the cost of increased power consumption. This could easily be experimented with. At least one full horizontal readout has to be performed to ensure the register is clean. On completion SRUN* only is forced active and the frame sync' signal, FSYNC*, being high enables the camera bitmap. The storage area clocks start, the vertical counter counts to zero, addressing the bitmap, which is sampled to determine what is to be done with the first row of the storage area (and succeeding rows). For windowed camera formats, the excess rows, before the window starts, have to be transferred out of the CCD. This is illustrated in the bitmap block description. With the horizontal clocks in the standby state, charge from the rows of the storage area is summed in the horizontal register. As this includes dark current as well as signal, during a long sequence of transfers the register may overflow and could contaminate rows yet to be read out of the storage area. In this case Dump row readouts of the horizontal register would be inserted in the sequence to flush the register.

Eventually, the first row of the window is reached. The vertical clocks stop, controlled by the bitmap output 'BMD1', and a horizontal readout starts. The design allows a two clock period settling time before the horizontal clocks change, and a single clock period for the horizontal clocks to settle before the verticals change. Throughout the length of the window, a single clock of the storage area alternates with a horizontal readout, which is triggered by SRUN* deasserting. Normally the two rows at the start

of the window will be dump rows to remove the charge built up in the vertical transfer sequence. The remaining rows hold valid window IDs and thus are used to collect events.

There are still rows remaining in the storage area and these must be removed via the horizontal register so SRUN* remains active while these are transferred to accummilate in the horizontal register. When the full 288 rows of the CCD have been removed, the vertical counter loads with a negative number (−291 set with pull up/down resistors on AY[1:8]), FSYNC* is asserted, the bitmap disabled and IRUN* asserted. This is the end of the storage area readout phase and the start of another frame transfer period. The signal 'VPL*' controls the load inputs of the counters — an AND gate allows the bitmap **MACS** bus access circuitry to set the address for the bitmap, using 'CTRLOAD*' when the camera logic is not active. Additional gating in this path selects the different ways of ending a frame — these are described within the bitmap section. The 'catch-all' signal here is provided by an 8 bit comparator on the Y counter outputs YC[1:8] (it is enabled when FSYNC* is high). This will cause the end of the readout phase actually after *287* rows have been transferred as the LSB, YC0, is not decoded for simplicity. CCD row 288 is swept away during frame transfer, and thus cannot be used. This is shown in Fig. 15. All of the ending controls are gated with 'VGATE'. Bitmap output 'BMD3' (when high) can be used to end the frame in the same way as the comparator.

One aspect of this module is the counting of vertical clocks. A programmable counter (which thus accommodates changes in vertical clock rate) divides the system clock by 6, providing two pulses per vertical clock period, the counter being held in the load state by SRUN* low. This is illustrated in Fig. 12. The first, 'VCLE', is provided by decoding the divider counter output, and increments the Y position counter which addresses the bitmap. The second, 'VGATE' (divider ripple count), after an access period is allowed, gates the output of the bitmap. The bitmap determines if the vertical clocks should stop at this point — the pulse is timed to allow this to occur when the row transfer has fully completed and the vertical clocks to the CCD are passing through the standby state. This gives convenient phasing between the bitmap output and the position in the CCD. For example, at row 2 of the CCD, the counter clocks to 2, the bitmap indicates whether the row now loading into the horizontal register is to be readout *and* holds the relevant window IDs for row 2.

The block also produces, by sampling the bitmap BMD0 output during VGATE, the signal, 'DUMP*', which is asserted for the full frame period except for the time when useful, not dump, rows are being transferred from the horizontal register. This can be used to enable the event processing circuitry and reduce power consumption. As BMD0 is high for a row to be readout if it contains data, DUMP* goes high in VGATE for BMD0 high and returns low when BMD0 is low, as shown in Fig. 14.

5.4 Camera bitmap

This block features a lookup table, or bitmap RAM, whose purpose is to control the frame readout sequence and to assign areas of CCD pixels to data windows. The circuitry is held on Camera 4 and Camera 5. The RAM used is 64K locations of 4 bit words, which is used to map to CCD pixels. The word size allows each CCD pixel to be allocated a number from 1–15. This is the Window ID number. Thus a block of locations in the RAM containing, say 6, indicate that the corresponding CCD pixels belong to data window 6 and that any events detected within these pixels will be assigned to data window 6. The window ID is passed to the link to the **DPU** with the event position in order that the **DPU** may sort the events into different areas based on the window ID. A window ID of zero is not allowed because it is taken as meaning that such a CCD pixel does not belong in any window and is merely being removed from the horizontal register. The RAM offers great flexibility in producing windows of any size or shape but does imply that no windows may overlap though can adjoin in X or Y.

As the RAM used is just 64K (to keep cost down etc), there are only 8 bits for X and 8 for Y to specify the location in the bitmap which corresponds to a CCD pixel. Nine bits in X and Y would be needed to point to individual pixels. Hence some restrictions are imposed. Of the X and Y position counters in the horizontal and vertical logic, the least significant bit is ignored and the next 8 bits passed onto the bitmap address lines (XC[1:8] and YC[1:8]). Thus pairs of pixels ('twixels') are addressed in both X and Y in give the smallest accessible unit of 4 CCD pixels. This allows more relaxed timing for the RAM. This also means that windows must start at *even* valued CCD pixels in X and that in Y, the window IDs start from even numbered CCD rows. More on requirements on Y later.

The X counter connects to the least significant portion of the bitmap RAM address — with not all of the RAM needed to specify windows on a row there are unused locations which can be used for other control purposes. Refering to the horizontal and vertical logic blocks it is found that on completion of a vertical transfer, the Y address of the bitmap is the row number/2, while the X address is 'Z'/2 — this is the location that must be programmed with information on what will be done with this row of the CCD as a whole. Actually the same action will be performed on a pair of rows. The defined codings are shown in table 4.

RAM value	action
0	No horizontal readout
2	Horizontal readout, but dump it
3	Horizontal readout, potential data
+8	Transfer and skip to frame transfer

Table 4: RAM codings for action on rows

The use of the bitmap in generating a camera format is ilustrated in Fig. 10, which also shows how the bitmap is programmed. The X address increases from left to right (0 being the first pixel out of the CCD appearing at the output of the data analysis array) and Y from top to bottom. The row type shows the coding of the action for the row pair. As shown, in addition there some windows, indicated by a block of the bitmap being filled with the same ID value. All other locations of rows that are read out and that are not dump rows, should be zero showing that they are not to be used for detecting events.

The first few rows accumilate in the horizontal register (A row action coding of zero is placed in the bitmap). Next this must be removed, before real data can be accepted, by a pair of readout and dump rows (coding RD). This is followed by a pair of full horizontal readout rows that have no window ID's set that account for the borders around a window (discussed shortly), then full readout rows containing window IDs. At rows 18—19 there is a gap in the windows, but since there is a window following immediately, these are readout fully and used for borders. Some more rows with window IDs follow and there is a larger gap to the final window. The border rows must be clean so they have to be preceeded by a pair of readout and dump rows, which takes care of this gap. If the gap were larger, then the excess rows would have a row action coding of zero to accummilate charge in the horizontal register. After the final window row is readout, the remaining rows of the storage area are transferred to the horizontal register. This example makes use of the terminate and goto frame transfer bit of the row action coding. This causes the readout sequence to skip to the start of the frame transfer period after it is received, saving a little time. The action of frame transfer causes any charge in the whole of the storage area to be accumilated in the horizontal register and so is equivalent to completing the series of vertical transfers only. This feature should not be used with active window IDs in a horizontal readout as only the first row of a pair would be readout in this case, thus giving a odd length window.

When programming the bitmap by filling in Window IDs, the window starts at an even numbered CCD row. However, if centroided data is collected, it is found that event centroids are recognised, as shown in Fig. 10, in an area offset in Y (only) by one row such that for centroided data (ie. checked by the event validate block), windows start at *odd* numbered rows. The window will start one CCD row before the first occurence of its window ID in the bitmap RAM. This is done to save some electronics — there is no need to delay the window IDs by a row to match the data emerging from the central row of the data analysis array. To centroid an event in Y, data from three consecutive rows is required, a centroid coming from the central row. The data analysis array must store the central row and the previous, so with the Window IDs appearing on the current row, the IDs here match with events in the previously readout row (see data analysis array). That is, the ID's match centroids from the previous row. There is another consequence — To centroid a single row there have to be two border rows which must contain valid data (the top border row is that deepest in the data analysis array). This applies to any window as shown in Fig. 10 (dashed boxes). The impact on the programing of the bitmap is to ensure that before each window there is a pair of valid rows being readout, with either that they have other window IDs in, or a specially added pair with no IDs on the rows (Fig. 10). Failure to observe this can result in corrupted fringes to windowed data.

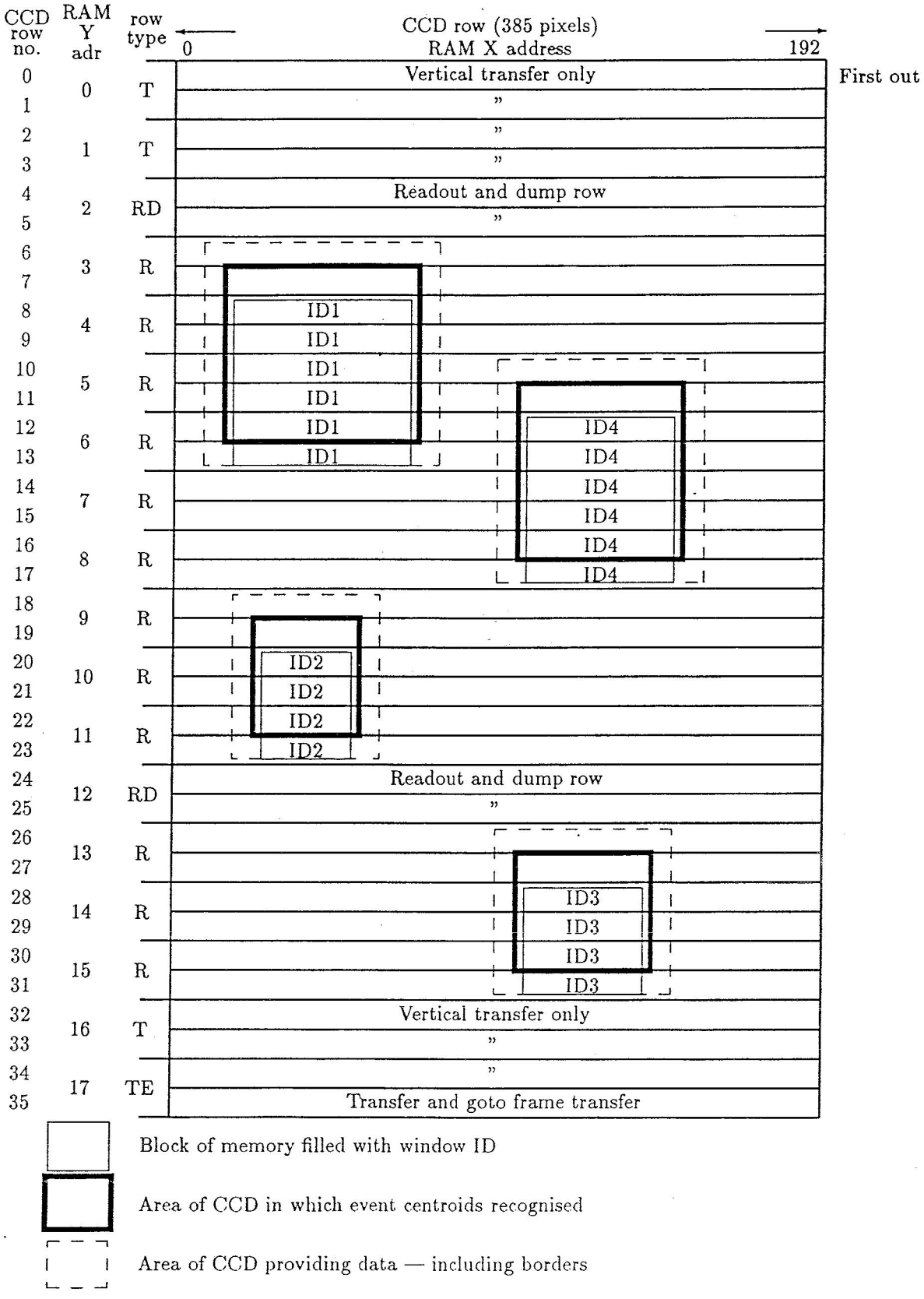

Figure 10: Using the bitmap to give a Camera format

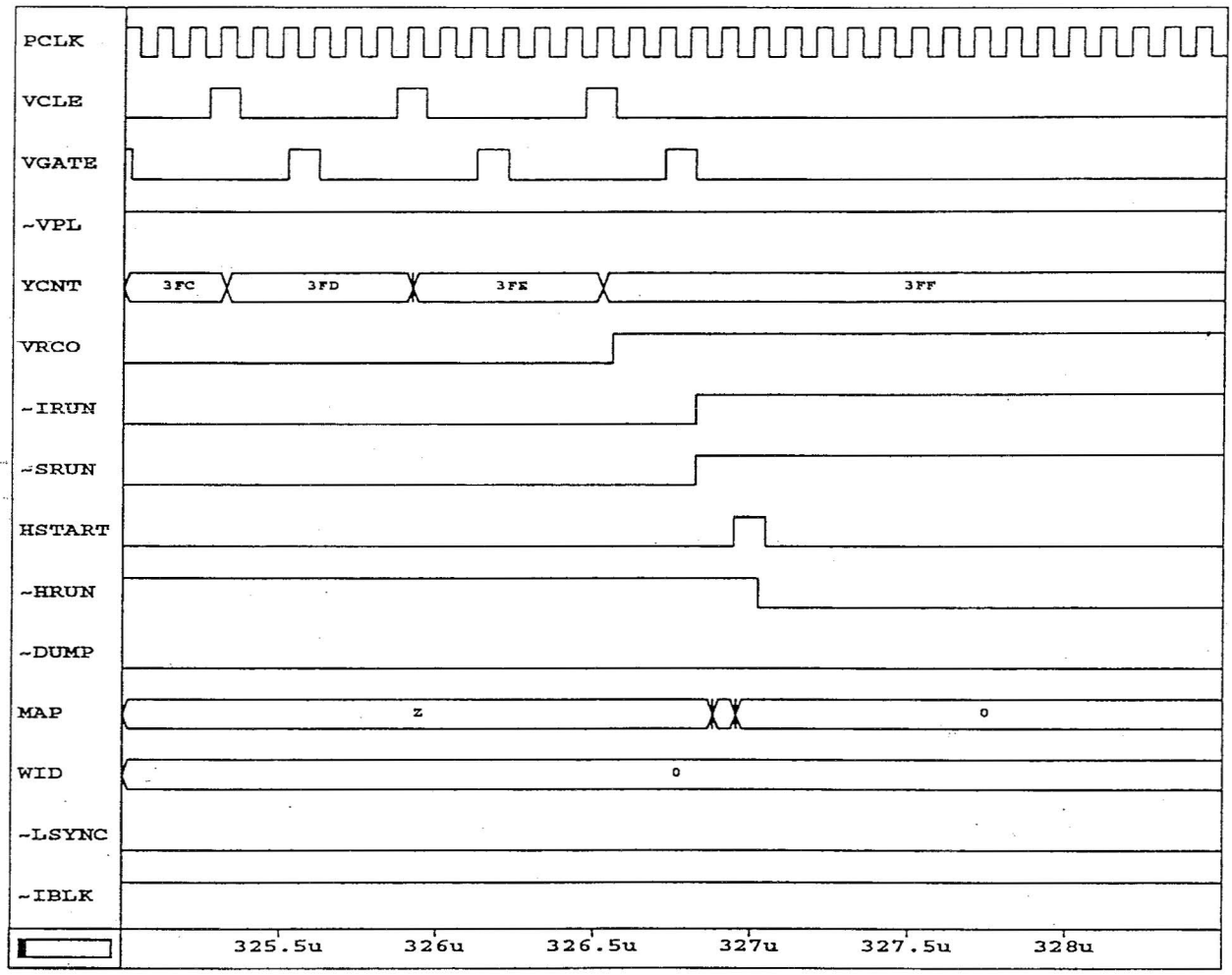

Figure 11: Timing for end of Frame Transfer phase.

Clearly there has to be border columns for the X centroiding to work cleanly as well, but these are taken care of automatically as all 385 pixels in a row are readout, are valid and can act as border pixels (windowed pixels can act as border pixels for another window etc). Any difficulties occur if a window were to start at $X < 2$ or window to end at greater than 382. The window ID values show which CCD pixels can hold valid centroided positions. The bitmap contents for $X > 384$ and $Y > 287$ (CCD pixels), apart from the row action coding, are ignored.

Full timing for an example of a CCD frame is shown in the following block of figures. Here the first 8 rows are transferred only, rows 8/9 are dump rows followed by 10 rows of readout, including two dummy rows. Fig. 11 shows the end of the frame transfer period and the start of an automatic dump row. During frame transfer the Y counter holds a negative value and IRUN* and SRUN* are both low as all CCD vertical clocks are synchronous — IRUN* asserted (or local reset 'LRESET*' low) disables the bitmap RAMs via their chip selects as they are not needed and power can be saved. LSYNC* is held low (Camera 6.) and DUMP*, so all event signals are invalidated (no window ID). As the CCD is transferring the 290th row the Y counter terminal count ('VRCO') is active — on completion VGATE causes both IRUN* and SRUN* to deassert (gating on Camera 7), halting all vertical clocks.

The rising edge of SRUN* automatically triggers a horizontal readout but, since DUMP* did not get a chance to go high as IRUN* was still low at the requisite clock edge in VGATE, this is a dummy readout with no black level sample nor Window ID's, despite the fact the bitmap is now enabled. This flushes charge built up in the CCD horizontal register during frame transfer. When finished, SRUN* only is asserted for the frame readout phase and as the vertical clocks start, VRCO goes low in the 'VCLE' period. The vertical counter is then zero so the bitmap can now output an row action code for the first

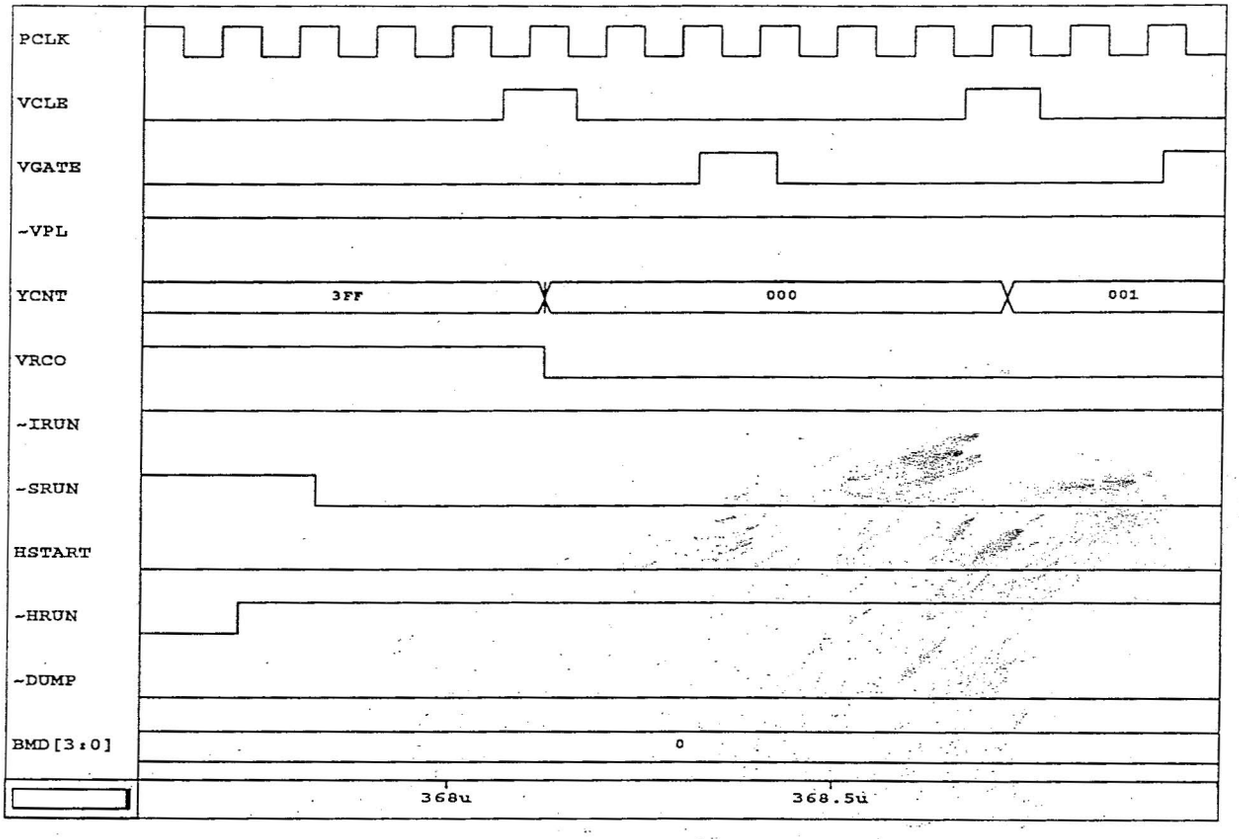

Figure 12: Timing for start of Storage area Readout (transfer only).

CCD row of the storage area (Fig. 12).

The first 8 rows are transferred into the horizontal register as Fig. 13 shows. During the VGATE period the bitmap action code is zero so the gating on the input of the SRUN* JK flip-flop from BMD1 keeps SRUN* active (Camera 7). As the 9th row is transferred, the bitmap outputs action code 2 for a dump pair of rows so SRUN* will be deasserted at the end of the vertical cycle.

The process of starting to readout a windowed area of rows is shown in Fig. 14. First there are two dump rows with vertical transfers alternating with horizontal readouts, then as CCD row 11 is being transferred, with a Y address of 10, the bitmap outputs a code of 3. During VGATE, BMD0 high cause DUMP* to go high (Camera 7), so Window ID's are activated for the row. However this row pair has 0 in all pixel bitmap locations so the resultant window ID's ID[0:3] (on Selector) remain zero — these are border rows for the window. From then on, there are non-zero locations in the bitmap and these indicate window ID's on the following rows. These are quickly indicated by the 'WIN ACT*' signal (Camera 6) — here there are two windows on a row. The row action code is 3 for all these rows. Eventually the last row of the windows is readout and the next vertical transfer gives an action code of zero so that DUMP* goes low in the VGATE period.

In this example there is nothing more in the frame so there is a choice in terminating the frame readout phase. In the first case, all the remaining rows of the CCD have their action codes zero so as illustrated in Fig. 15, a comparator on YC[1:8] produces 'LASTROW' during the transfer of the 287th CCD row. During VGATE, 'VPL*' is produced which is connected to the load inputs of the vertical counter so that it preloads with -291. As it does so gating (Camera 7) causes IRUN* to assert to join SRUN* which has been continuously asserted. A new frame transfer phase starts. Note the short period of the -291 state — an early vertical clock explains why -290 is not used. In the second case, Fig. 16, the early termination action code is used after a number of vertical transfers and implies there will be an odd number of rows readout of the CCD storage area (the first of a pair of rows is used). Bitmap output BMD3 is used just like the comparator output, as long as IRUN* high permits. The third case should not be used as Fig. 17 shows — if the early termination code is combined with a horizontal readout, then the 2nd row of

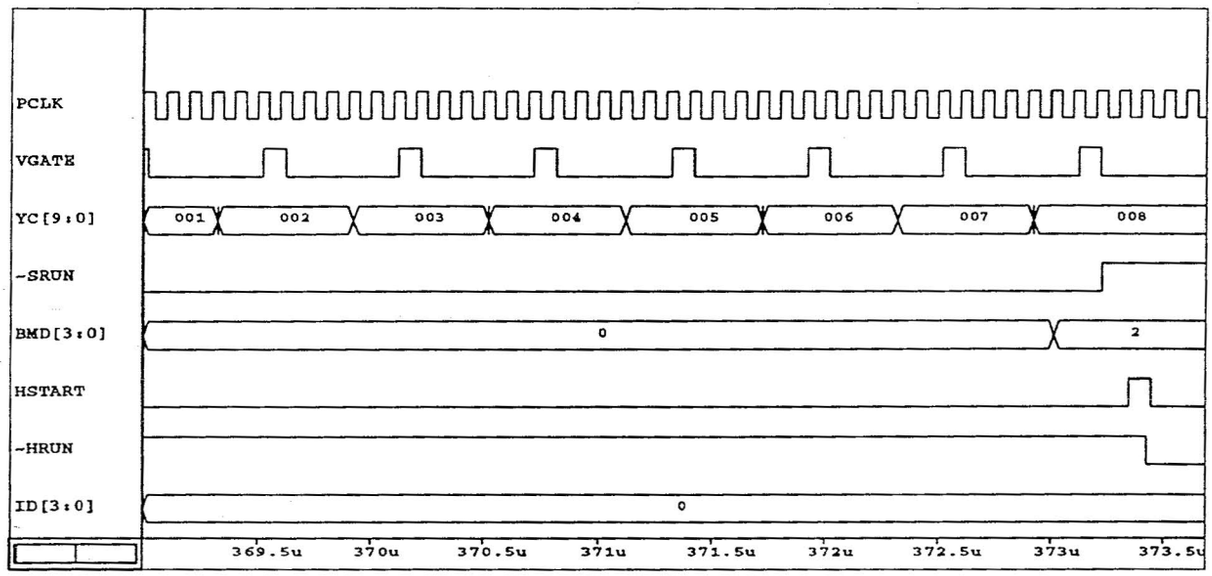

Figure 13: Storage area transfer only Timing.

the pair is not readout having gone to the frame transfer phase, resulting in an odd length window. The terminate code is best used on its own.

The programming of the bitmap is done via the **MACS**bus by first loading the address of a location into a 16bit counter/latch. The 8 MSB bits being the address of the Y row pair and the LSB belonging to X. The value to be loaded into the bitmap is written to another **MACS**bus sub-address, which causes it to be written into the RAM and the X address of the RAM to be incremented (not Y), by enabling the X counter for two clock cycles. The bitmap contents can be read back in the same way. The bitmap cannot be updated whilst the camera is running as there are access time conflicts between the camera and **MACS**bus interface — indeed the counter/latches are used by the camera as the X and Y CCD position. This means that to update the bitmap, the camera must be halted by placing it in a local reset state with another command (the line 'OC CMODE*') over the **MACS**bus, where no timing signals are generated, the CCD sits in a standby state and the centroiding logic is inhibited. There is then free access to the bitmap RAM. In this state a latch (Camera 4) generates the 'LRESET*' signal. A second latch holds a single bit value from the data bus MBD[0:15] strobed in by 'OC CMODE*'. After 3 clock cycles its value is in the LRESET latch. The delay and synchronisation is important on startup. It is unimportant where the CCD sequence is halted. This could also be considered a low power state, and is enforced by Power On Reset.

Once the CCD is halted, the bitmap loading timing is shown in the example of Fig. 18. The address information from the **MACS**bus is held in transparent tri-state latches connected to 8 inputs, excluding the LSB, of the X and Y counters. A value from the data bus is held in them on the rising edge of the strobe 'OC BADR*' allowing the RAM update to complete in its own time. In the computer access mode these latches override the normal preload input values set by pull up/down resistors, for the counters, and the counter control signals from the normal operating logic are inhibited. The bitmap access strobes are disabled by 'LRESET*' high as protection against disturbance by invalid **MACS** bus cycles. The address strobe is sampled at PCLK rate by a D-type to provide the 'CTRLOAD*' passed to the counter load inputs so that the counters output the RAM address on the next clock edge(s) (also shows why transparent latches are used).

With an address set up, the data write strobe, 'OC BDAT*', is issued with the data bus holding the value to be written to the RAM. The RAM data inputs are connected directly to the bus. The protected strobe (Camera 5) drives the RAM write enables so a value is written directly. Read and write strobes are ORed to provide the single signal 'BACC*' indicating a data access — with further gating on Camera 5 it operates the chip selects of the RAMs which would otherwise be disabled. The completion of an access clocks a D-type (Camera 4) to generate a two clock cycle signal 'XINC*' enabling the X counter in order that its RAM address can auto-increment by two. This allows a bitmap row to be quickly loaded

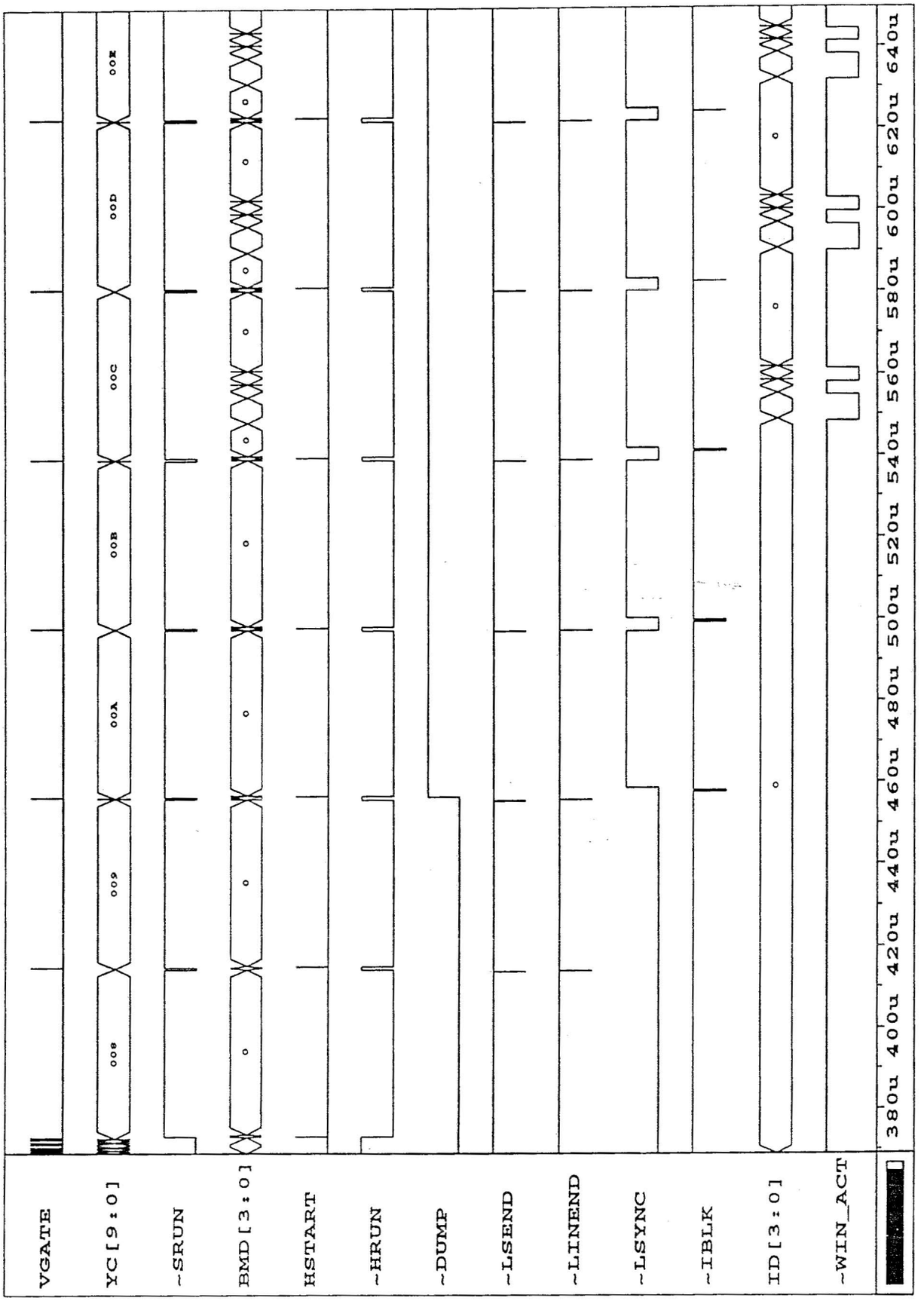

Figure 14: General Timing for reading out a data Window.

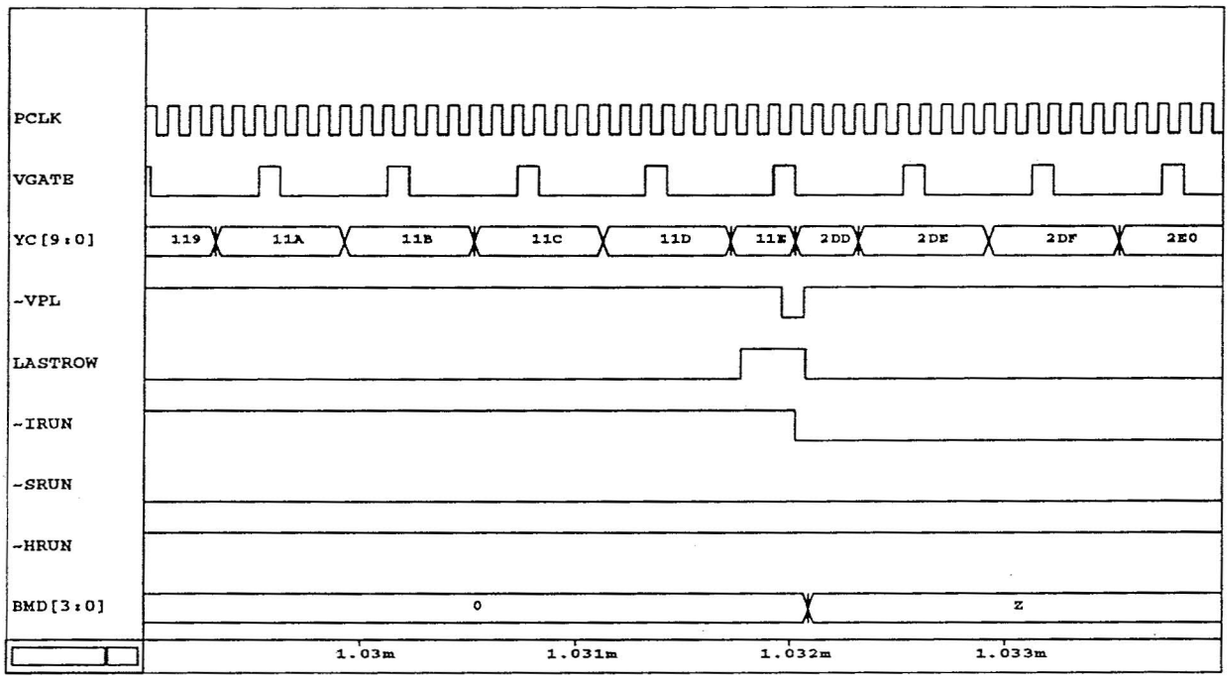

Figure 15: End of frame timing — to last CCD row.

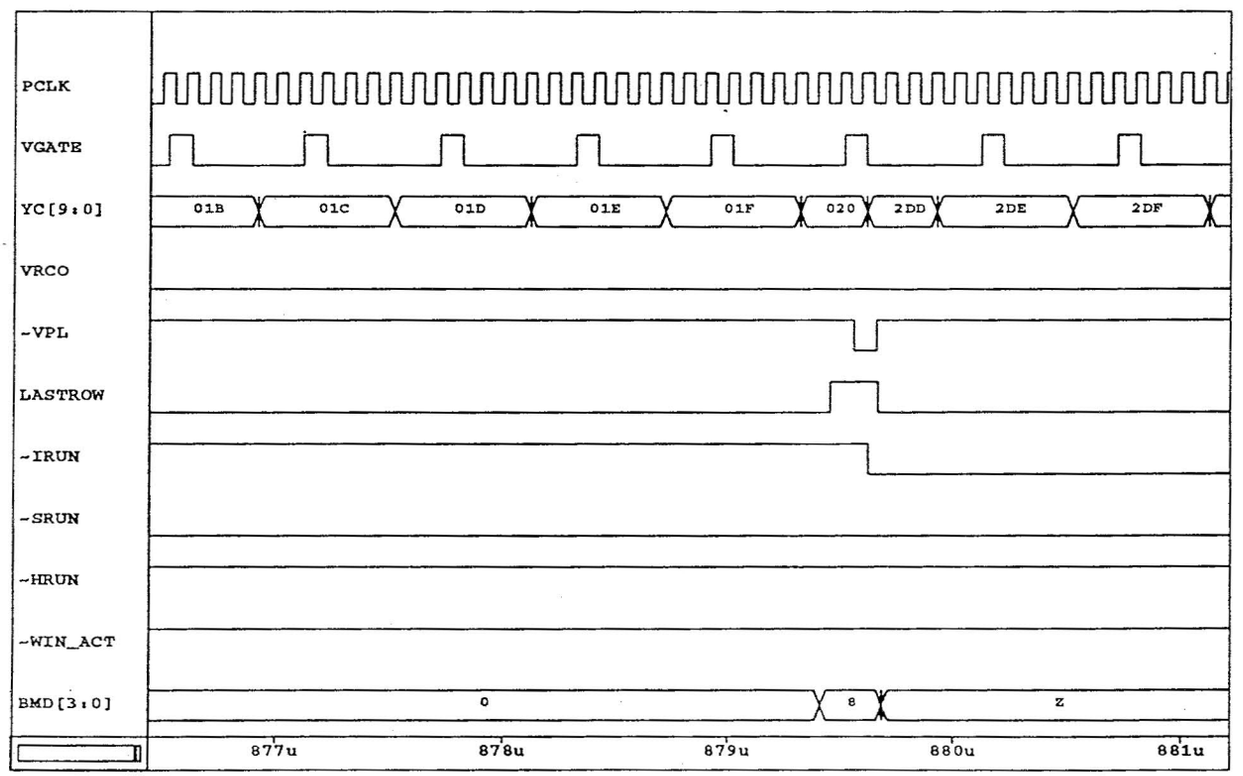

Figure 16: End of frame timing — early terminate code.

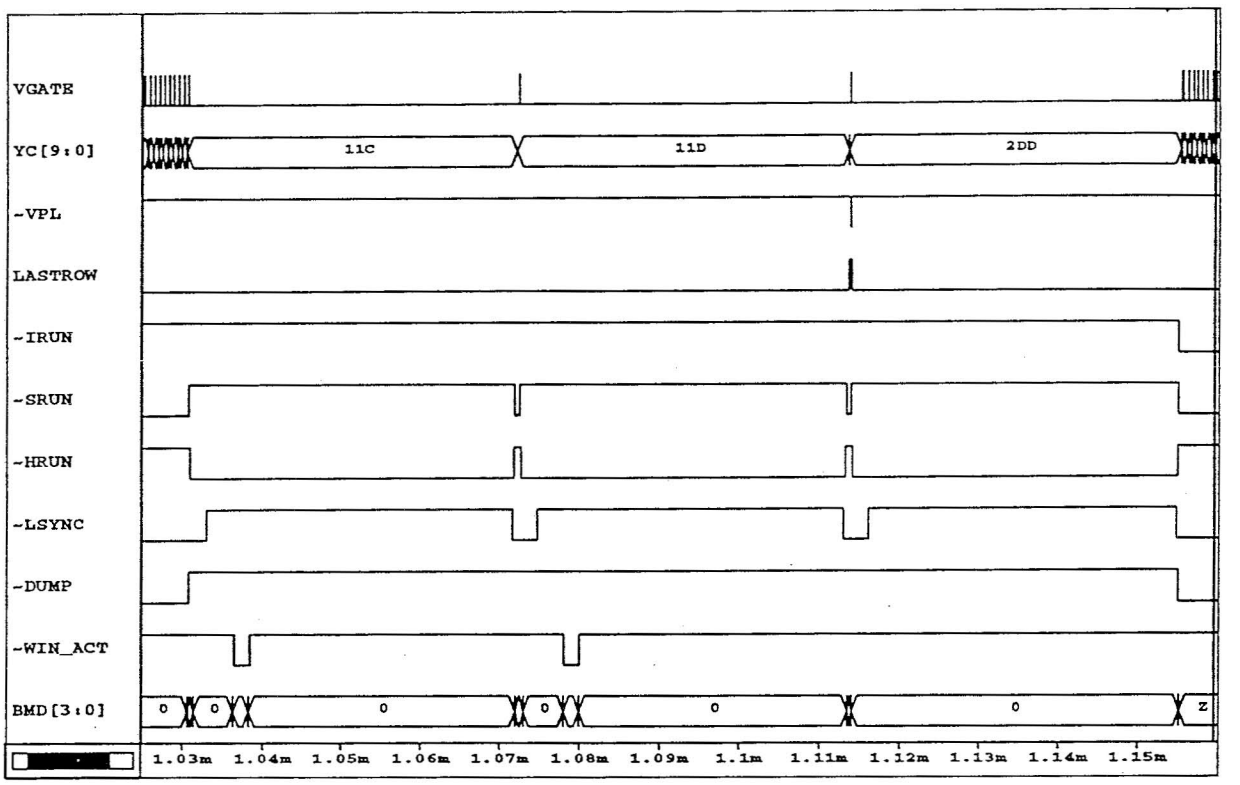

Figure 17: End of frame timing — with horizontal readout.

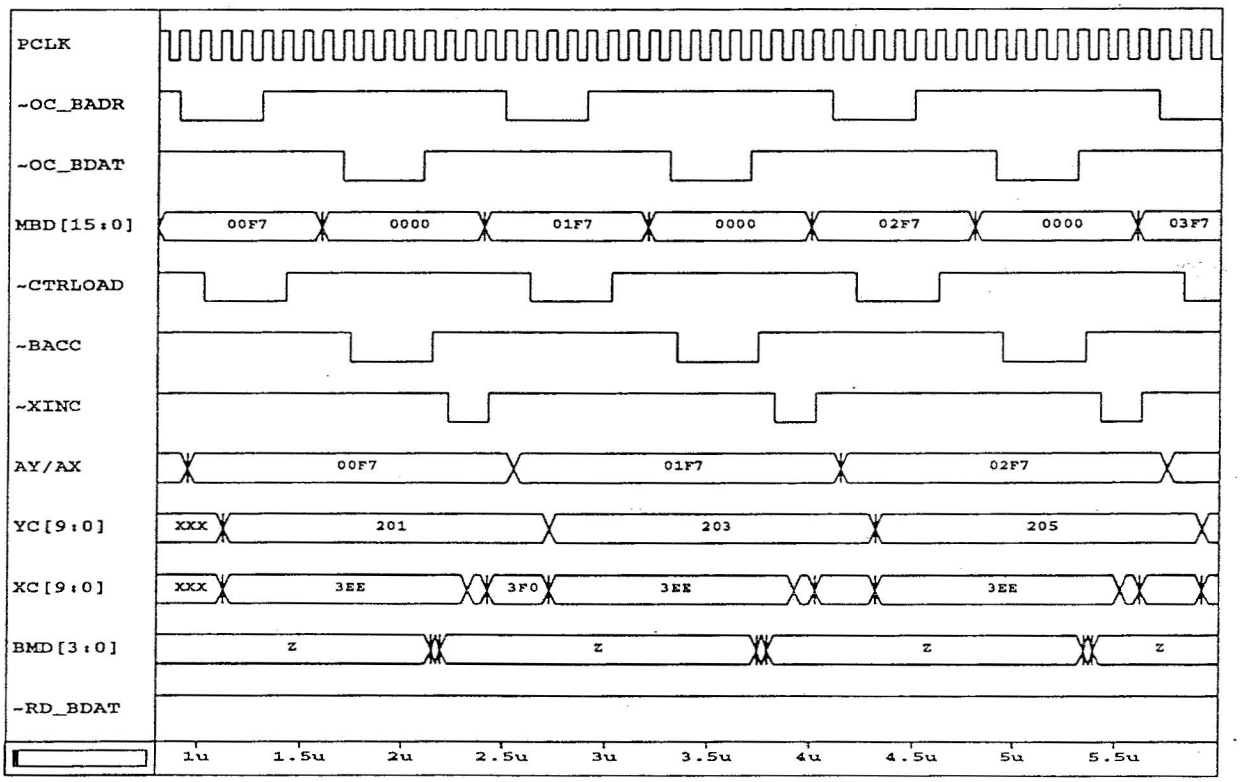

Figure 18: Timing for loading the Camera Bitmap RAM.

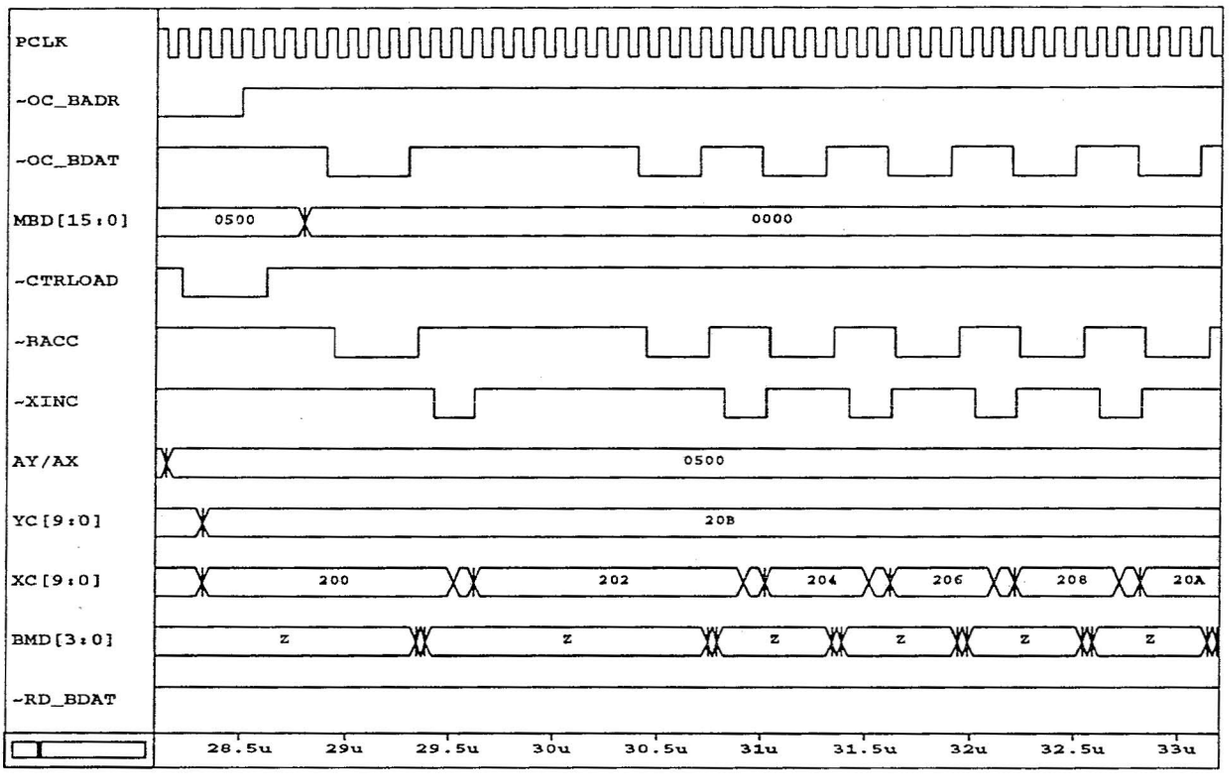

Figure 19: Bitmap access auto-increment feature.

as only the starting address need be loaded — Fig. 19 shows this feature more clearly. Reading back of the bitmap contents is similar, with the read strobe, 'RD BDAT*' being used instead of the write strobe — it enables a tri-state buffer to allow the RAM outputs out onto the least significant 4 bits only of the MACS data bus. Fig. 20 illustrates.

When programming is complete the local reset state is removed using 'OC CMODE*'. On the next edge of PCLK, NOR gates (Camera 4) force CTRLOAD* low until LRESET* returns high a further two cycles later so that the X and Y counters are reloaded with their normal values in readiness for the first frame Then the camera restarts from the beginning of the frame transfer phase. This logic is shown in Fig. 21.

6 Data analysis array and event validation: Darray card.

6.1 Black level subtraction

This circuit block is placed with the array as it prepares the digitised data stream for use by the data analysis array. The raw digitised CCD data from the ADC in the detector head sits on a bias level (black level) which varies gradually from row to row of the CCD and this needs to be removed in order that event thresholding on a baseline of zero can work. As described in the camera horizontal logic block, there are 4 pixels per line that can be used as a sample of the black level — the horizontal logic provides a signal (IBLK*) to indicate these.

With the horizontal clocks halted, during a vertical transfer, a 10 bit accumilator (implemented with counter devices) is cleared (Darray 9). The accumilator output, DARK[0:9], is connected to one port of a adder, with the incoming 8 LSB of the 9 bit CCD video data (VD[8:0]), synchronised by a latch to the system clock (PCLK), on the other. It is assumed that the black level can be measured with 8 bits. The timing is shown in Fig. 22. During IBLK* active, the accumilator is synchronously loaded with the adder result, so that on completion of IBLK* the accumilator holds the sum of the four reference pixels. Should the 8 bit adder overflow, its carry output causes the counter producing the 2 MSBs of the total to increment, thus dealing with the situation where the mean black level is greater than 64. The

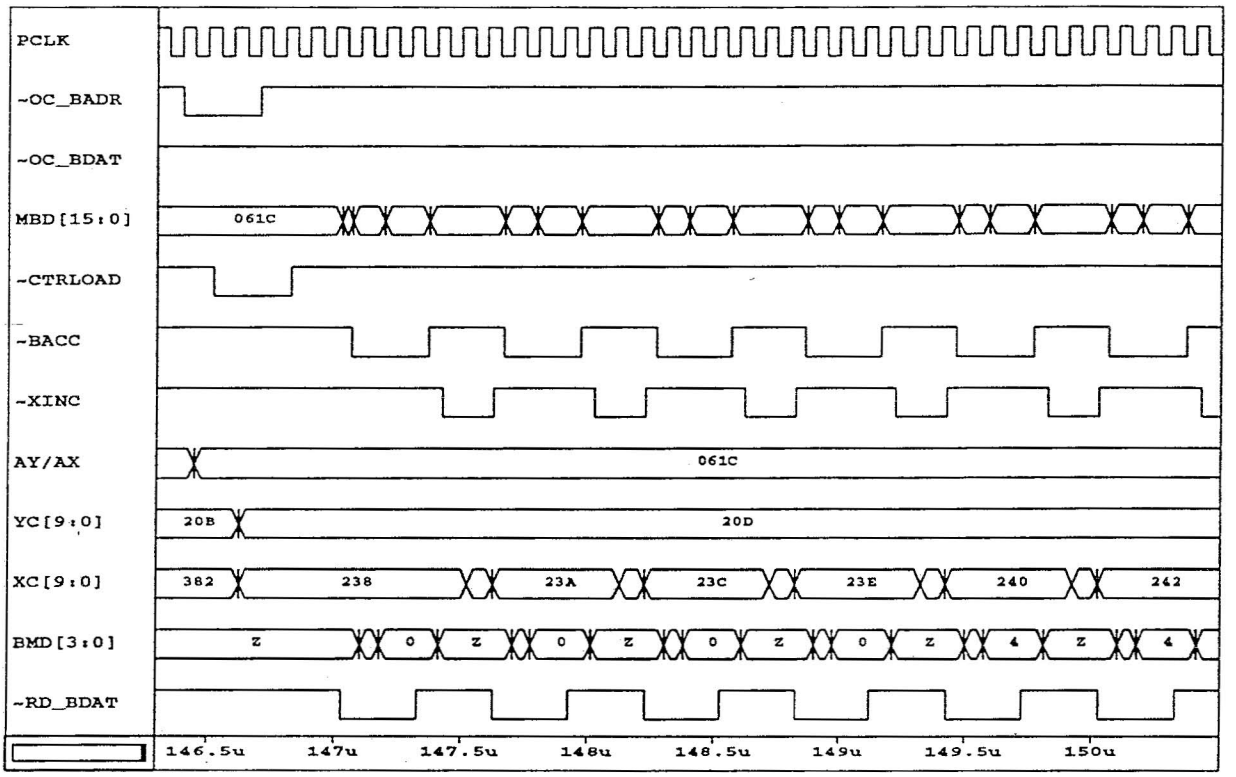

Figure 20: Timing for read back of the Camera Bitmap RAM.

most significant 8 bits represent the mean value — with a value of two added, using the X counter least significant bit, XC0, during the IBLK* period, this will be the rounded up/down value. For example, a level of 62.75 will be rounded to 63.

A second 8 bit adder with latch (Darray 10) can then subtract the mean black level from the subsequent pixels of the row. The 8 bit adder used processes DARK[9:2] and the least significant 8 bits of video data. Because of the additional delay through the octal inverter, there is insufficient setup time for the result of the first subtraction after IBLK*. However, for EEV CCDs this belongs to a transition element and won't be used in a window. Since a pixel of data plus dark level may exceed 8 bits (but not 9), the most significant video bit needs to be considered at this stage. The carry of this 8 bit adder is latched to give a ninth bit (SUB CA) and the video bit VD8 is delayed to match (VDB8). If both are high then the pixel value exceeds 8 bits — the latch output is disabled and the pull-up resistors give a saturated value of 255, which is better than wrapping round to a low value. On the next clock cycle the final 8 bit counter/latch loads the final data pixel value to give DAA[7:0]. Should the result of the subtraction of the black level be negative, the clear input of the counters is held low for the cycle, forcing the final pixel value to 0 instead of wrapping around to an anomalously high level. The result is one of; subtracted data, zero or saturation. Also, there is a pipeline delay of two clock cycles once the incoming video data is synchronised to the system clock.

6.2 Data analysis array

The detector head interface presents a serial stream of CCD pixel data from the current row. The modules that process that data need simultaneous access to data from the 2 rows previously read out of the CCD. The purpose of the data analysis array is to provide 3 serial streams of data from consecutive rows that are in phase. ie. pixel 3 of rows 4, 5 and 6 appears at the output at the same time. The key to this are two storage elements capable of storing all 407 elements of a horizontal readout, regardless of whether they will all be used for event detection and centroiding. Simplistically, these may be considered shift registers delaying the serial data stream by a whole line. The action of the data analysis array is illustrated in Fig. 23.

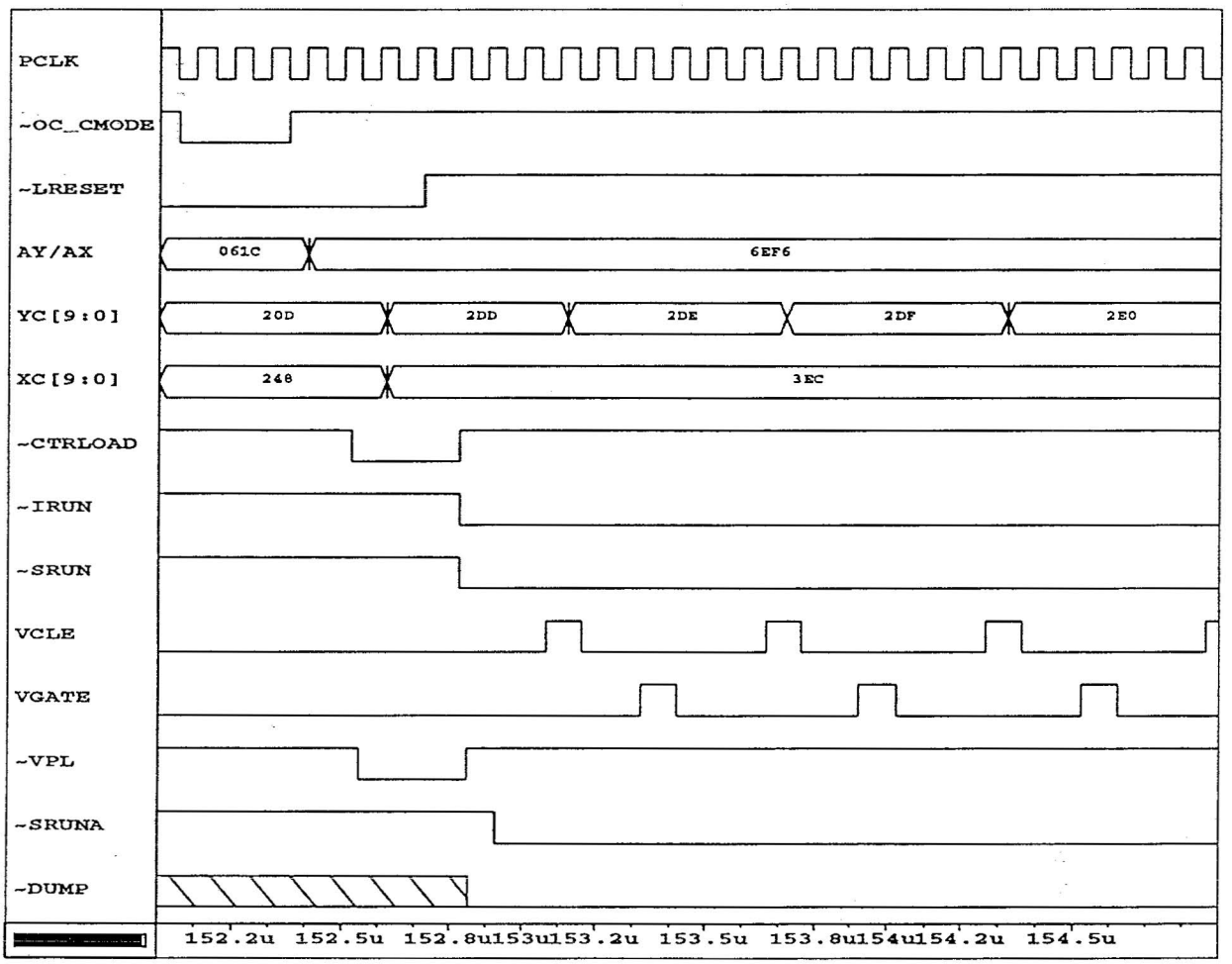

Figure 21: Starting the Camera readout sequence.

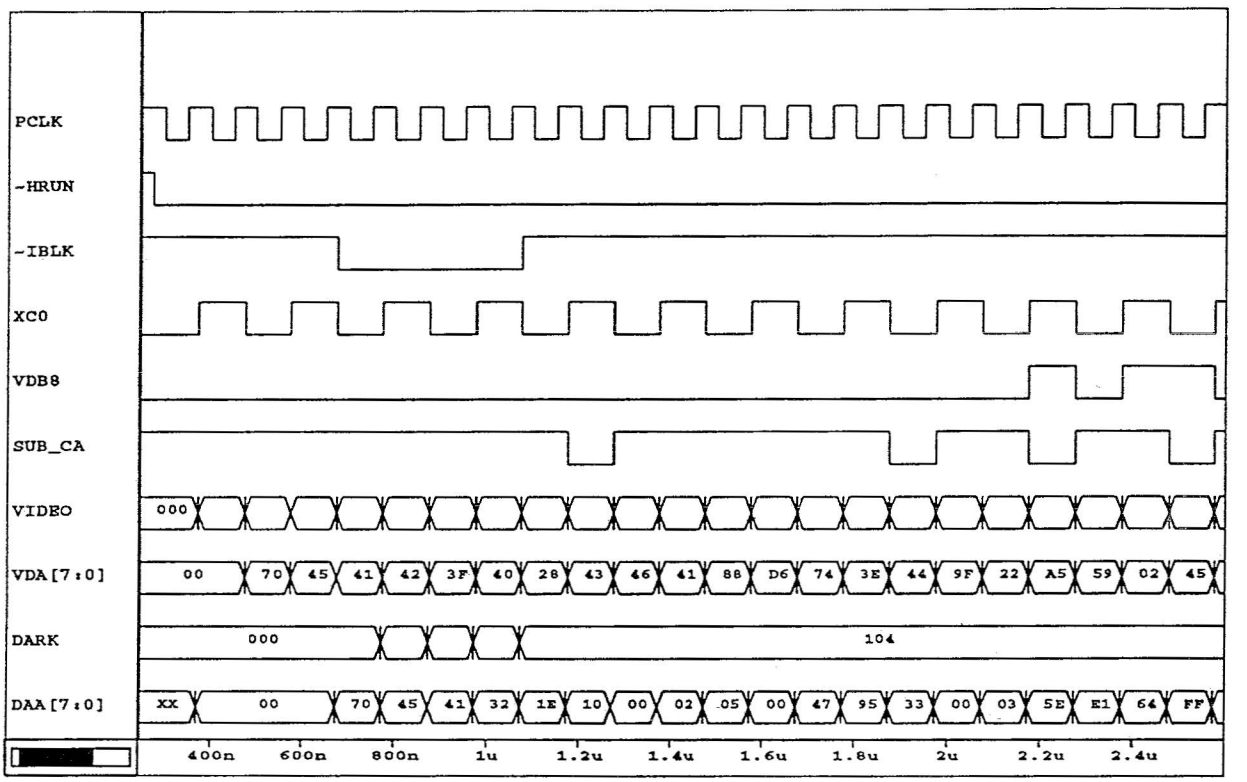

Figure 22: Timing for Black Level subtraction.

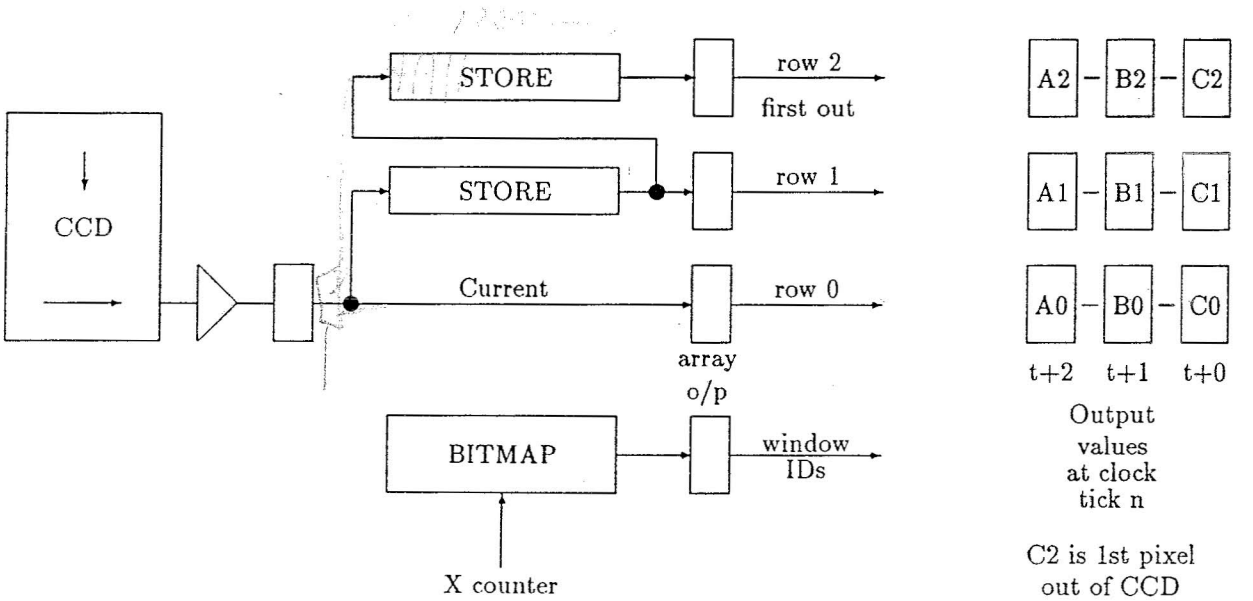

Figure 23: Function of the data analysis array

There are three output ports of which row 2, which is the deepest in the storage units, corresponds to the row which was first clocked out of the CCD. Window IDs appear in parallel with data from row 0 which is the current row from the CCD, but they refer to centroided data from row 1. A centroid in Y is calculated about a centre in row 1. To produce a two dimensional window, through which all the CCD data flows, for centroiding in X as well, 'time' is used. At clock tick 't+0', the column 'C', first out of the CCD, appears at the output of the array, followed at 't+1' by the central column 'B' and the next 'A' at 't+2'. The columns are not required simultaneously as careful pipelining in the succeeeding processing stages can pick out the required elements. Element 'B1' from Fig. 23 is defined as holding the centre of an event when it is correctly centroided so its 'appearance' at the array output is used as a timing reference point.

This design implements the storage elements in FIFOs — there are two sets used, each device being 512 words deep. Because each device requires an access period (2 clocks) followed by a recovery period, three interleaved devices are needed per row. ie. a device stores/recovers every third CCD pixel value. These are shown in Darray 3–5. For writing into the first row of FIFOs an 8 bit counter/latch captures a digitised pixel value and holds it for use by its associated one of three FIFOs. They are controlled by a 3 output ring counter that runs whilst HRUN* is active and ensures the FIFOs are in phase from row to row (Darray 6). Timing for the analysis array is shown in Fig. 24 (first row) and Fig. 25 (second row). This ring counter is formed with an octal latch and gating to provide one clock cycle pulses OE0*, OE1* and OE2*. These are coupled to the Load inputs of the counters. The captured data values for the three columns are F0I[0:7] to F2I[0:7]. It is not important which of the initial data values gets captured first — they are rubbish anyway — as long as it's the same one each row.

The FIFO write pulses (2 clock periods long), W0* to W2*, are generated by a second ring counter (Darray 6) again using the octal latch and gating, that runs on HRUN* active, and the row not being a dump row (DUMP* deasserted), so the analysis array is active for horizontal readouts containing true data only. Data valid on F0I etc, at the rising edge of W0* is written into the FIFO giving ample time for data hold and write recovery — an interleave of less than two would require dealing with time units less than one PCLK period.

The second ring counter also generates FIFO read pulses, R0* to R2*, in phase with the write, so each FIFO is simultaneously read from and written to. The FIFO outputs are R1F0[0:7] to R1F2[0:7] with pull-down resistors. These resistors ensure the buffers are not floating when the FIFO output is tri-stated so consuming power. Multiplexed buffers using the OE0* to OE2* signals, select a FIFO output value (F1BUS[0:7]) which is latched as the row output. The current row (0) has 2 latches in series so its data is in phase with the other rows, countering the one pixel delay in writing to the FIFO. The final row (2) is similar except that the timing permits direct connection of the FIFO inputs to the outputs of the corresponding device in the previous row. Different signal names are used of course. The final three parallel data streams are D0R[0:7] (current), D1R[0:7] and D2R[0:7] and these are used by the parallel processing blocks. The diagrams were produced with a row length of 50 pixels so the final data streams are seen as '50 apart'. For example, at 23.2u we have 02H, 34H and 66H.

It is important that to maintain an exact row delay, each row should have exactly the same number of reads and writes from each FIFO. The row delay needs to be set up in the FIFOs. All the FIFOs are reset during the frame transfer phase and start the frame empty — there is no delay between data in and out. Timing for this operation is shown in Fig. 26. A control flip-flop is cleared on IRUN* low (Darray 6), inhibiting reads, using OR gating, from the FIFOs after it has been resampled by PCLK to avoid runt pulses. Now the array waits in an inactive state until the DUMP* input goes high at the start of a row. For this CCD row the FIFO write pulses only are enabled so that both rows of FIFOs are prefilled with 407 pixels of data. Only the first row gets sensible data at first — the second gets zeros as there are no R0* etc pulses. Gating on Darray 6 requires DUMP* high when a VGATE is present. Thus (DUMP* changes after VGATE) for the next CCD row the prefill flip-flop is set giving the signal 'FIRSTD'. Now reading from the FIFOs is enabled. The second row of FIFOs outputs zeros while the first border data row is written into it. From the row after, normal operation of the array is resumed to the end of the frame and by the time the border row pair is passed, the array is presenting the data rows in the correct order, whereupon the window IDs control the acceptance of events.

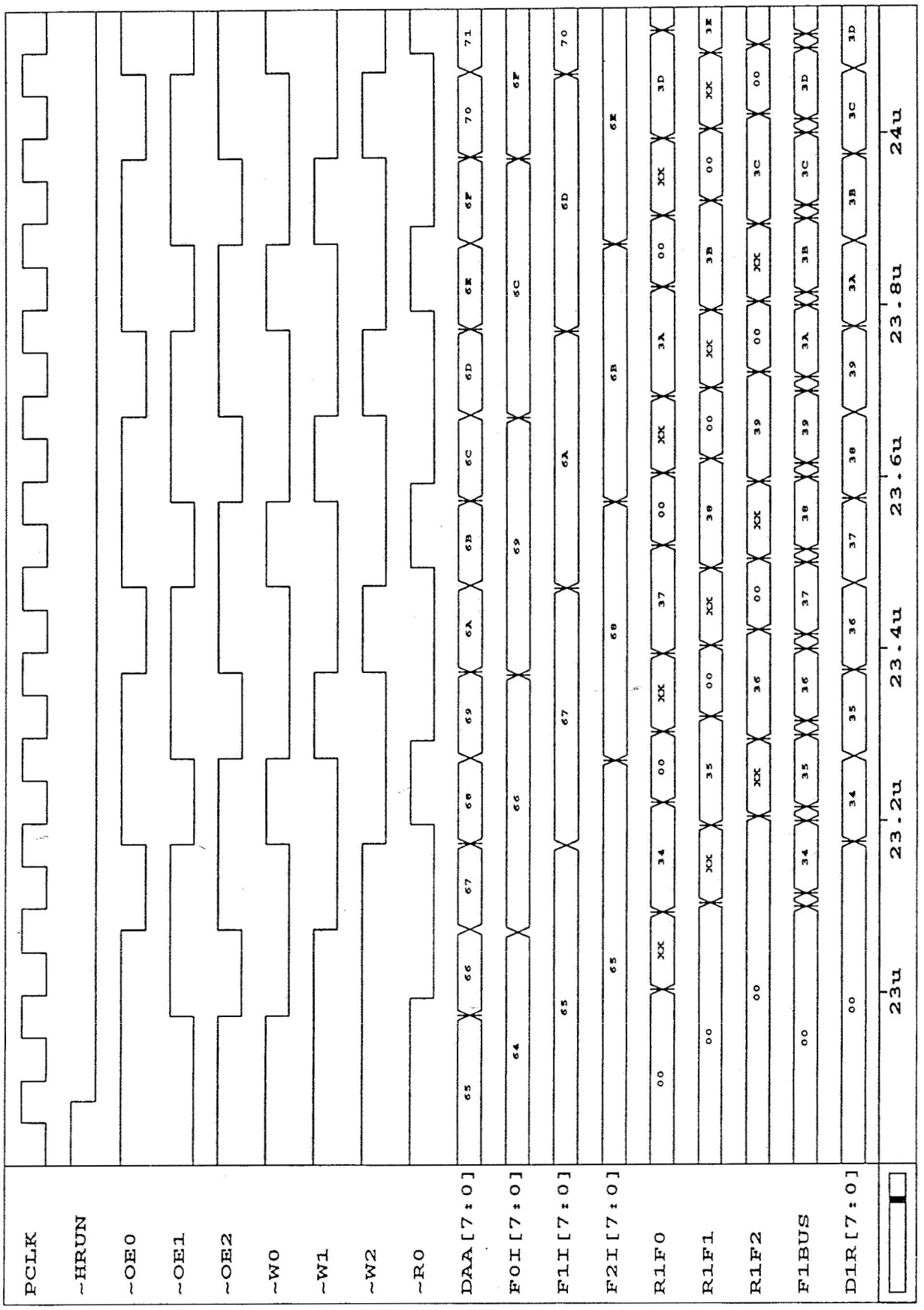

Figure 24: Timing for the Data analysis array (first row).

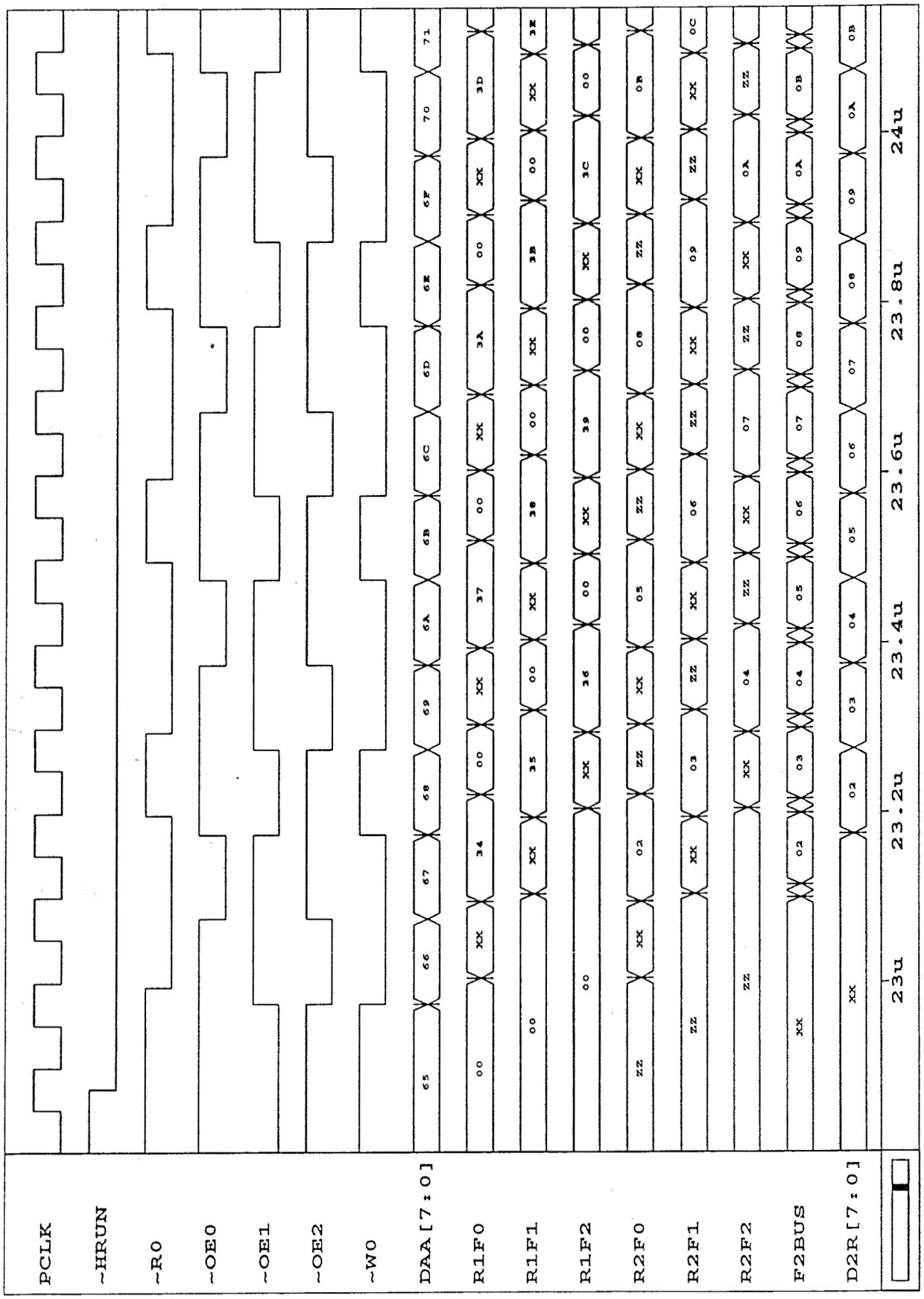

Figure 25: Timing for the Data analysis array (second row).

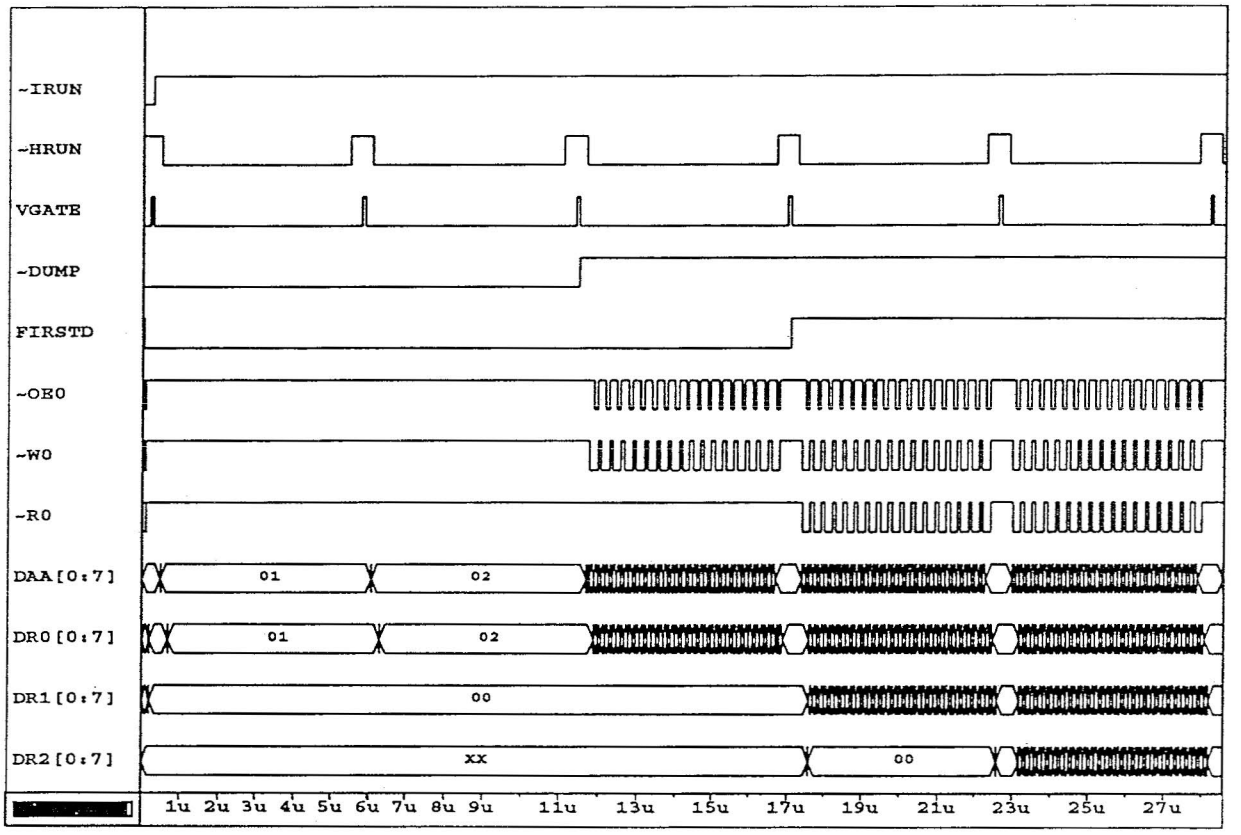

Figure 26: Prefilling the Data analysis array.

6.3 Test port

This tiny unit sits on the output of the analysis array and allows connections to facilities not provided in the Blue Module electronics such as a frame grabber and real-time display. These are very useful test and evaluation features and are invaluable for debugging the whole detector head and camera logic. These facilities exist in EOB-1, for example, and could be used by connecting up to the camera interface port of EOB-1, providing 8 video data and 4 timing lines (26 way connector). Fortunately, all these signals are naturally produced within the Blue Module camera logic and there are *no* need of additional electronics, just a 26 way header plug to connect to.

6.4 Event validate

This block examines the output of the data analysis array to determine whether a valid event is currently centred here, that is, in element 'B1' from Fig. 23. If there is one, an ECD (event centre detect) signal is generated to act as a strobe for the output of other logic blocks. As it is a dynamic process, with data running continuously through, a stream of ECD signals could be produced, one per event. The circuitry is shown on Darray 7–8.

With the aid of time a 'cross-hair' of elements is examined for peaks in the data. So using the element nomemclature from Fig. 23, the following criteria must be met to produce a possible 'ECD'; $A1 < B1 \geq C1$ and $B0 < B1 \geq B2$ and $B1 > threshold$.

On centroiding the event, finding its peak position *within* a CCD pixel, the assumption is made that it is always within element B1, which is true for all but infrequent, peculiar (asymmetric) events. The conditions imply that in units of a CCD pixel, taking an origin at the centre of a pixel, the centroid range is; $-0.5 \leq centroid < 0.5$. C1 and B2 (first in data stream) are to the '−ve' side of B1. The $\geq$ condition picks up events with two pixel values equal whilst the $<$ condition ensures there is a gap between events (or dip). Events cannot be detected at less than 2 CCD pixels apart giving a maximum peak event rate

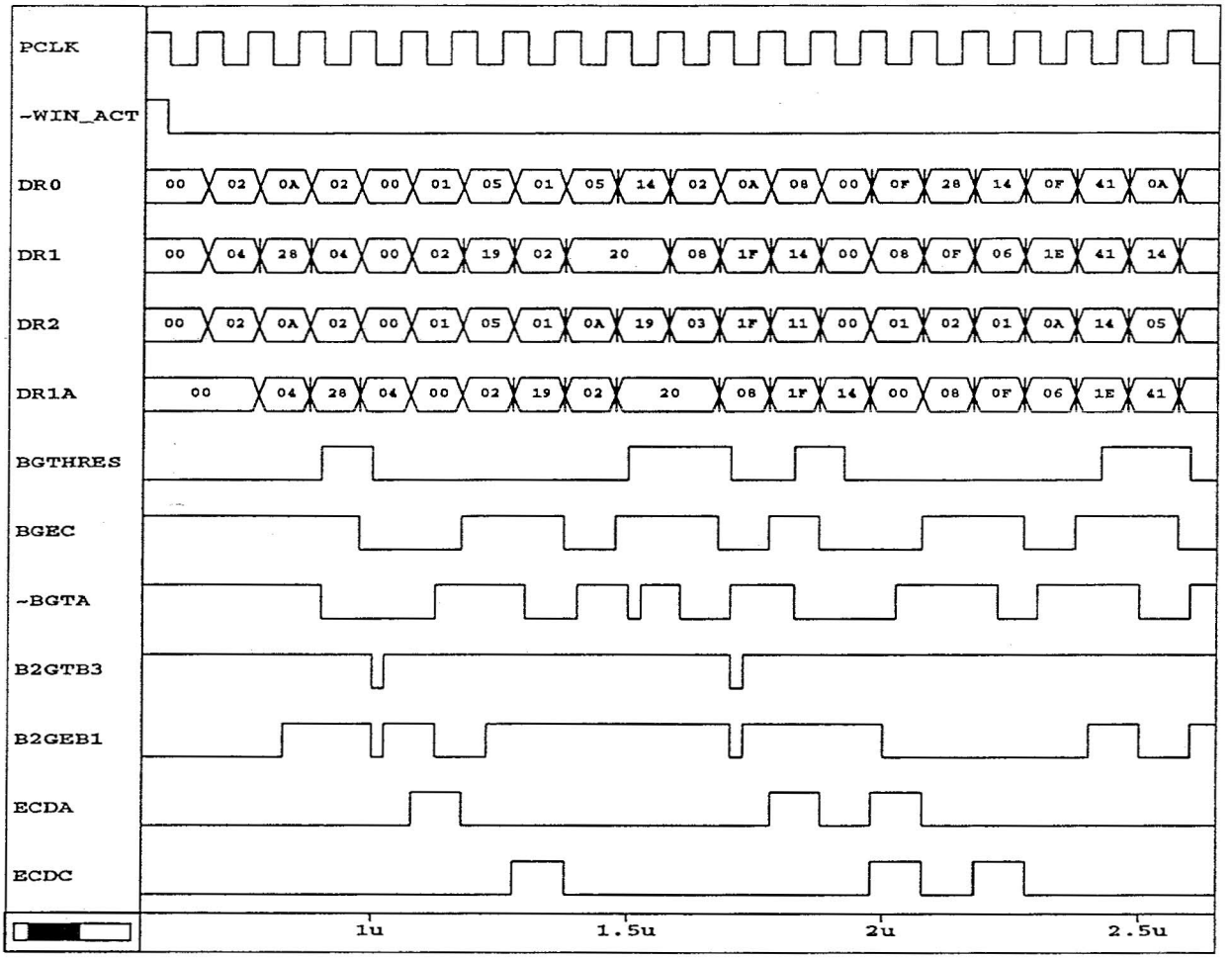

Figure 27: Timing for validation of an event.

of 5MHz (important for the FIFO buffer). The comparison of the event peak against a threshold value ensures that there is sufficient signal to noise to centroid the event and reject noise spikes. This value is programmable (8 bits) off the **MACS** bus, using the strobe 'OC PTHRES*'. This value, represented by 'PTHRES[0:7]', should be the two's complement of the threshold required as it saves an inversion for a subtraction operation.

This section is implemented using pipelined adders which act as comparators with one port of 8 bit data complimented. Timing is shown in Fig. 27. The X comparisons are done with one 'comparator' saving the result of the first in a latch on Darray 8 (the first comparison is active low BGTA* and the second high, BGEC). After all the comparsion results are latched/pipe-lined, they are gated together on Darray 8. There is a final qualification stage. The window active signal from the bitmap logic, after a delay to match the pipelining delay through the validation, is gated with the comparison result to produce the final ECD signal. Thus only events that occur within the windows of the camera format will be accepted. A shift register allows the ECD to be tapped off at various points to strobe other processing blocks or logic, depending on the amount of pipelining required. These are 'ECDA', 'ECDC' and 'ECDE'. The earliest ECD signal is present 3 clocks after an event is centred on the output port of the analysis array (element B1). The signal 'COMPACC*' clears all ECD signals when the centroid lookup tables are being accessed from the **MACS** bus to prevent false events.

7 Event processing and centroiding: Process card.

7.1 Event energy and Double counting

This block uses the output of the data analysis array in parallel with other processing blocks, such as event validate, finding the sum of the 3 by 3 pixel values centred on each B1. Thus in each clock tick, $energy = (A0+A1+A2)+(B0+B1+B2)+(C0+C1+C2)$. This event energy is available for measuring. To fully represent the energy, the summation would have to be performed to 12 bits ($9 * 255$). However for the final value we can drop, say 2, least significant bits to use an 8 bit value plus an overflow bit. This can only be done at the final step. It is important to take note of the most significant bits, otherwise the 8 bit value can 'wrap-around' to give an apparent low value. Summing this number of elements is expensive to terms of adder devices and the amount of pipelining required. Indeed, in this design it is the slowest process and all stages (eg. the FIFOs) that are dependant on it must wait. It takes 5 clock cycles from an event centred on the analysis array output port to present the energy (and a further 2 cycles to indicate a Double count). The bulk of the circuitry required is shown in Process 10–11.

The energy event is used as input to the FIFOs. It is also used as input to the Double count logic. A typical single event within the 3 by 3 area (analysis window) may give a maximum energy of 500 A-D units. Another event occurring on top of it, or nearly so, will raise the energy considerably. If the event energy is compared against a threshold (say 600 ADU) and is found to be greater than a flag is set. This 'Double count' flag indicates there is more than one event at this centroid position and may be used to add a count of two to that point in an integrated image. It is not particularly reliable as there are problems with the process, such as the event(s) centroid(s) not being accurate in this case. An 8 bit comparator, using adders, in this design compares the energy, less the 2 LSB, against a software programmable threshold, taking into account the overflow bit, and sets a flag latch if exceeded. The feature can be disabled under software control, holding the flag inactive.

The calculations involved here are shown in Table 5 — this shows the 3 parallel data streams from the analysis array and the intermediate results. The underlined numbers track the progress of an event from it being centred in the array to when it produces the Double flag (all numbers in decimal), each line being a system clock tick. Refer to the circuit diagrams for the intermediate signals.

The outer two CCD rows are summed on Process 3 to give an 8 bit result, 'D0+2R[0:7]', plus carry, 'D0+2R8' (after a latch). Another adder on Process 10 sums this with D1R[0:7] to produce one pipeline delay later the 8 bit column sum 'COLSUM[0:7]'. Eventually during the same clock cycle 2 MSBs, COLSUM[8:9], appear at the MSB unit of an adder and its carry, which is used to add carries, D0+2R8 and intermediate result, together. Now it gets more complex — three column sums must be added for the total. The first column sum formed (first out of CCD) is saved in a latch and then added to the current sum to thus form, after latching the addition, the sum of the first two columns of the event, 'COLB+C[0:7]'. The pipeline latch ensures that a final adder, and latch, will add in the new (3rd) current column sum to produce the least significant part of the total energy, but this is not the full result — at this stage the least significant two bits of the energy are dropped. A parallel chain must handle the carries from all the addition stages in order to produce the more significant bits of the energy. A similar technique is used on the carries COLSUM8–9, using the carry outputs of the lower data bits, at the correct time, as a source for the most significant adder(s) carry in inputs. The final addition stage produces the 4 most significant bits of the total energy — two of these are latched with the lower bits to produce the result, 'E[2:9]', while the most significant are ORed together to produce an overflow signal, 'EOVF', which is active if more than 10 bits are needed for the sum.

A comparator, formed from adder devices, compares E[2:9] against a latched threshold value, 'ET[2:9]'. This value, which is strobed in by the signal 'OC ETHRES*', should be the twos complement of the required value (this is 500 in the simulation example). The signal 'ETHRES' is ORed with EOVF on Process 8 to create 'DOUB' which in turn is latched to produce the active high double count flag, 'DOUBLE' (Process 11). Thus, overflow events would always exceed the threshold. A second control latch captures MACS bus data MBD15, using 'OC ETHRES*', to give an enable line 'DOUBLEEN' — if high, double events will be flagged otherwise DOUBLE will be held low. The default state on power up is for double events to be disabled.

It is possible to operate the event validate based on exceeding a threshold in event energy and not the

D0R (7:0)	D1R (7:0)	D2R (7:0)	Colsum (9:0)	ColB+C (10:0)	E (9:2)	Ethres	doub	double
0	0	0	x	x	x	x	x	x
10	80	30	x	x	x	x	x	x
0	0	0	0	x	x	x	x	x
0	20	0	120	x	x	x	x	x
0	90	0	0	120	x	x	x	x
0	50	0	20	120	x	x	x	x
0	0	0	90	20	30	0	x	x
1	10	1	50	110	35	0	0	x
10	100	10	0	140	27	0	0	0
1	10	1	12	50	40	0	0	0
0	0	0	120	12	35	0	0	0
2	15	9	12	132	15	0	0	0
12	82	21	0	132	33	0	0	0
2	15	9	26	12	36	0	0	0
2	12	2	115	26	33	0	0	0
15	82	15	26	141	9	0	0	0
9	15	9	16	141	35	0	0	0
0	8	0	112	42	41	0	0	0
9	45	3	33	128	39	0	0	0
31	96	10	8	145	38	0	0	0
3	19	1	57	41	40	0	0	0
0	0	0	137	65	38	0	0	0
40	70	30	23	194	24	0	0	0
80	180	50	0	160	50	0	0	0
30	50	10	140	23	54	0	0	0
70	100	40	310	140	40	0	0	0
31	65	23	90	450	40	0	0	0
90	200	100	210	400	112	0	0	0
19	46	29	119	300	135	1	0	0
1	5	2	390	329	152	1	1	0
60	120	75	94	509	104	0	1	1
120	255	255	8	484	179	1	0	1
90	150	130	255	102	150	1	1	0
5	7	3	630	263	123	0	1	1
7	220	10	370	885	89	0	0	1

Table 5: Event Energy Calculations.

peak pixel value. However, since we must wait some time for the event energy, the validation block will be held up and all the other processing blocks that need an ECD signal would have to wait longer. In practise, this needs extensive use of delay latch chains.

7.2 X Centroid

The function of this block is to calculate the centroid of the data on the output of the data analysis array in the X direction. If we assume an event is centred on the array output, then centroiding implies measuring the central position of the event profile to an accuracy of a fraction of a CCD pixel — the event validate section determines this to the level of one pixel. The centroiding blocks actually run continuously, but the result is only valid in conjunction with an event validate signal, which can be considered an enable or mask control. The triangle or parabola algorithm is used:

$$centroid = \frac{A1 - C1}{2(2B1 - (A1 + C1))}$$

This gives a result $-0.5 \leq centroid < 0.5$ but the designs (eg. MIC ground based) use a version giving the range $-1.0 \leq centroid < 1.0$.

$$centroid = \frac{A1 - C1}{2B1 - (A1 + C1)}$$

The different range is handled by the lookup table programming which in effect calculates the actual division. This removes a multiplication by 2 in the hardware which actually creates no further accuracy as the LSB would always be zero.

With the aid of suitable delays and adders the design is able to produce two numbers **m** and **n**. These are[2]:

$$m = A1 - C1$$

$$n = 2B1 - A1 - C1$$

Here **n** is unsigned and calculated initially to 9 bits accuracy while **m** is signed and also calculated to signed 9 bits accuracy. The division is difficult to do in hardware and so a lookup table is used to do this process utilising this input. The lookup tables are actually RAMs as described in the lookup table section. To keep the tables within a practical size, 64K RAMs are used allocating 8 bits to **m** and 8 bits to **n**. The values calculated so far must be reduced in size in a sensible manner, still maintaining centroiding accuracy.

For normal, single events the peak height (B1) is 75 on average, with a maximum of 140 approx. The events are fairly narrow so the surrounding pixels, A1 and C1, are much less, say 20 if the event was in the middle of a pixel. Here **n** cannot overflow even 8 bits and taking the worst case of A1=C1=0, B1 must exceed 127 to overflow 8 bits with **n**. Taking **m**, to exceed a signed 8 bit value, the difference in A1 and C1 must exceed 127. The worst case (remember A1 and C1 must be less than B1) is A1=0 or C1=0 and C1=127 or A1=127. This would be from a 'two pixel' event of brighter than average intensity, in itself rather unusual. Hence it appears that for normal, single events a 8 bit range in **m** and **n** would be exceeded infrequently and full centroiding accuracy obtained with the 8 least significant bits.

An 8 bit range will be frequently exceeded for unusual events such as those that are asymmetric or, most often, large events due to the effect of coincidences — these may even have triggered the Double Count logic. Since the maximum value of a pixel is 255, a nine bit value for **n** or signed **m** can never be exceeded. One could use drop the least significant bit, using the other 8, but this could throw away information.

The method used in EOB-1 was to use the most significant 8 bits of **n**, dropping the LSB[3]. The least significant signed 8 bits of **m** were used but limited, that is, if **m** greater than 127 it is held at 127 while if **m** less than −128 it is held at −128. A side effect was the range changed to $-2.0 \leq centroid < 2.0$. The limiting is necessary to prevent an effective sign swap. For example, if **m** were found to be −134,

[2] Centroiding uses the single pixel values and not the sum of rows or columns that some previous centroiding schemes have used

[3] A earlier version used the least significant 8 bits of **n** that were forced to 255 if n was greater than 255. This prevents the lookup table input wrapping around to a small number — a large error.

the truncated 8 bit value would be +122. This is a major error whilst the limiting has a much more gentle effect on the data but even so, is only brought in for large events or multiple events. As events get bigger, the centroid position tends towards zero. If the centroid range is exceeded the pathological event will at least be placed on the correct side of the range.

The MIC ground based program introduced a new technique which is incorporated into this design — autoranging. For the smaller events both **m** and **n**'s least significant 8 bits are passed unchanged to preserve the fullest accuracy. However, if the 8 bit range of *either* **m** or **n** is exceeded, then *both* **m** and **n** are divided by two, passing on the most significant 8 bits. The centroid range ($-1.0 \leq centroid < 1.0$) is preserved so the process is invisible to the lookup tables. This actually uses less circuitry than limiting the result. The only effect is the loss of one bit's worth of accuracy. Loss of centroiding accuracy only affects the results if the centroid is close to the boundary between one sub-pixel and the next (binning of the centroid range — refer to the lookup table section), so the maximum error is one sub-pixel. The maximum error occurs for **m** at 128, where **n** must be greater than 128 (remember the range size). Considering combinations of 128 and 129, then the effect of truncating the LSB is: $-1/128 \leq centroid\ error < 1/129$. The maximum error thus occurs at the edges of a CCD pixel. Also this maximum error implies a large event (here B1> 128). If, for example we always truncated the LSB, the maximum error would be $\pm 1/40$! In comparison we could consider the effect of CCD noise, and the error due to digitisation ($\pm 1/80$ max). In conclusion, the autoranging offers the best compromise — the small errors will occur for unusual events and for a few large 'normal' events (depends on pulse height distribution). The unusual events frequently arise due to the effects of coincidence on the CCD, so positional accuracy has already been lost and hence a small loss of centroiding accuracy is irrelevant. There will be no gross errors due to sign changes etc.

In hardware, the auto-ranging uses a 2 way selector controlled by the carry from the **n** value or a signed overflow generated from the XOR of the carry and most significant bit of **m**. The circuitry is shown in the diagrams Process 7–9. The pipeline delay from the analysis array to the correct **m**-**n** values is 5 clocks. The process of calculating a centroid from the 3 parallel streams of data is shown in the simulation run of Table 6 (at system clock rising edge), and the effect of pipelining should be apparent. The input data stream is D1R[7:0] — the first 3 columns are decimal, the remainder Hex'. The progress of a typical event is shown by the underlined figures — the event being considered centroidable when its peak, '90', appears in the data stream.

Sheet Process 7 produces the X **m** value 'XMD[0:7]' from the D1R[0:7] data stream, using suitably delayed data to tap off the 'A' and 'C' values. The 9th bit, the carry of the adder, 'XM-C', is latched on Process 8 to match. The extra octal latch in the **m** chain ensures that it arrives at the same time as **n** — the 8th bit before this is available as 'XMC7' for sign calculation.

Sheet Process 8 calulates the X **n** value 'XND[0:7]' in two stages from D1R[0:7]. First the sum, A1+C1 (A+C[0:7]) is found using data values two clock ticks apart (the carry is also latched to give 'A+C8'). The delay chain is implemented on Process 3. These are inverted and added in two steps (gives the minimum chip count and because of insufficient time to invert and add 8 bits in a clock tick) to D1RB[0:7], now the peak value 'B'. The least significant bit of the adder is tied low so producing 2B as the algorithm demands. The first stage, with carry is latched together with the other half of the inverted byte and added in the next clock tick to D1RC[3:7]. Latching this gives XND[0:7]. The carry from the 8th bit is added to the inverted A+C8 plus D1RC7 to give 'XN-OVF' which is high if the 8 bit range is exceeded.

The auto-ranging for X is performed on Process 9. The two-to-one multiplexers select either the 8 least significant bits of XND and XMD or the 8 most significant bits on command (line 'X-OVF*' is low to select the most significant bits). This is either from 'XN-OVFD*', the n overflow, or 'XM-OVFD*'. The latter is generated on Process 8 where the XOR of XMC7 and the latched carry of the subtraction is the source. Its inversion provides the most significant bit XMD8. The simulation examples should make the operation more clear. The 16 bit auto-ranged result is latched to give 'MX[0:7]' and 'NX[0:7]'— these are available to the Lookup tables and can be tri-stated using the line 'COMPACC*' to allow down-loading of the tables.

7.3 Y centroid

The function of this block is to calculate the centroid of the data on the output of the data analysis array in the Y direction. Assuming an event is centred on the array output, then centroiding implies

D1R (7:0)	D1RA (7:0)	D1A+C (7:0)	XN-OVFD*	XM-OVFD*	X-OVF*	XMD (8:0)	XND (8:0)	MNX (15:0)
0	x	x	x	x	x	x	x	x
80	0	x	x	x	x	x	x	x
0	80	x	x	x	x	x	x	x
20	0	0	x	x	x	x	x	x
90	20	100	x	x	x	x	x	x
50	90	90	1	1	1	0	0A0	x
0	50	70	0	1	0	1C4	19C	00A0
10	0	90	0	1	0	05A	1CE	E2CE
100	10	60	1	1	1	01E	06E	2DE7
10	100	100	1	1	1	1A6	00A	1E6E
0	10	20	0	1	0	1D8	1C4	A60A
15	0	100	0	1	0	064	1B0	ECE2
82	15	25	1	1	1	0	0B4	32D8
15	82	82	0	1	0	19C	1B0	00B4
12	15	30	0	1	0	005	1E7	CED8
82	12	94	0	1	0	052	1CC	02F3
15	82	97	1	1	1	0	086	29E6
8	15	27	0	1	0	1BA	1C0	0086
45	8	90	0	1	0	043	1B7	DDE0
96	45	60	1	1	1	003	089	21DB
19	96	104	0	1	0	1B6	1C4	0389
0	19	64	0	1	0	01E	1D4	DBE2
70	0	96	0	1	0	058	1F2	0FEA
180	70	89	1	1	1	1E6	080	2CF9
50	180	180	0	1	0	1A0	1C6	E680
100	50	120	0	1	0	033	1A7	D0E3
65	100	24	0	0	0	0B4	1D8	19D3
200	65	115	1	1	1	1EC	0F0	5AEC
46	200	44	0	1	0	1B0	14C	ECF0
5	46	111	1	1	1	00F	055	D8A6

Table 6: X Centroiding Calculations.

D2R (7:0)	D1R (7:0)	D0R (7:0)	YN-OVFD*	YM-OVFD*	Y-OVF*	YMD (8:0)	YND (8:0)	MNY (15:0)
0	0	0	x	x	x	x	x	x
30	80	10	x	x	x	x	x	x
0	0	0	x	x	x	x	x	x
0	20	0	x	x	x	x	x	x
0	90	0	1	1	1	0	0	x
0	50	0	1	1	1	1EC	078	0
0	0	0	1	1	1	0	0	EC78
1	10	1	1	1	1	0	028	0
10	100	10	1	1	1	0	0B4	0028
1	10	1	1	1	1	0	064	00B4
0	0	0	1	1	1	0	0	0064
9	15	2	1	1	1	0	012	0
21	82	12	1	1	1	0	0B4	0012
9	15	2	1	1	1	0	012	00B4
2	12	2	1	1	1	0	0	0012
15	82	15	1	1	1	1F9	013	0
9	15	9	1	1	1	1F7	083	F913
0	8	0	1	1	1	1F9	013	F783
3	45	9	1	1	1	0	014	F913
10	96	31	1	1	1	0	086	0014
1	19	3	1	1	1	0	00C	0086
0	0	0	1	1	1	0	010	000C
30	70	40	1	1	1	006	04E	0010
50	180	80	1	1	1	015	097	064E
10	50	30	1	1	1	002	022	1597
40	100	70	1	1	1	0	0	0222
23	65	31	1	1	1	00A	046	0
100	200	90	1	1	1	01E	0E6	0A46
29	46	19	1	1	1	014	03C	1EE6
2	5	1	1	1	1	01E	05A	143C

Table 7: Y Centroiding Calculations.

measuring the central position of the event profile in a similar fashion to that in X. The triangle or parabola algorithm is used:

$$centroid = \frac{B0 - B2}{2B1 - (B2 + B0)}$$

The range is handled by the lookup table programming which in effect calculates the actual division as in X. Though the circuitry is slightly different due to the arrival time of the pixel elements, exactly the same method of finding **m** and **n** is used as for the X centroid blocks, and the same autoranging technique is employed. Both circuit blocks work independently but in parallel on the output of the data analysis array. The circuitry is shown on sheets Process 4–6.

The Y centroiding calculations are shown in the simulation run of Table 7. D0R to D2R are the output streams of the analysis array and the underlined numbers (input in decimal, outputs in Hex') track the progress of an event from it being centred in the array to the result which is fed to the Lookup tables. Each line is a system clock tick. Note, D2R is the first row out of the CCD.

Sheet Process 4 contains the calculation D0R - D2R giving a result 'D0-2RC[0:7]' plus carry. On sheet Process 5, D0+2R[0:7] (Process 3) is inverted and added to D1RB[0:7], left shifted with the least significant bit zero. The result is 'YNC[0:7]'. As for X, overflow signals 'YM-OVFD*' and YMD8 are produced here. Process 6 has an additional pipeline delay, in order to match up with the X result, which gives the YND[0:7] and YMD[0:7] results to feed the autoranging multiplexer. This operates as for X, though independently, under command of the 'Y-OVF*' signal. The selected value is latched (tri-stateable) to

provide a 16 bit address to the Y Lookup table (NY[0:7] and MY[0:7]).

8 Centroiding Lookup tables: Lookup card.

The function of this block is to produce from the outputs of the X and Y centroid blocks a position of the event within a CCD pixel. The range of the centroid position is divided into 8 bins known as sub-pixels. Ideally the boundaries of the sub-pixels are equally spaced as in; $-1.0 \leq$sub-pix' $0< -0.75$, $-0.75 \leq$sub-pix' $1< -0.5$ and so on to $0.75 \leq$sub-pix' $7< 1.0$. The assignment of events to sub-pixel number is done within these centroiding lookup table RAMs so in fact the actual sub-pixel boundaries are variable. There are two tables, one for X and one for Y, which are independent in that their contents are different giving different boundaries for X and Y. Things are arranged so that an event moving in increasing sub-pixel number also moves in increasing CCD pixel number.

The Lookup table RAMs are each 64K by 3 bits wide and have 16 bit counter/latches connected to the address lines. The 8 bit value **m** from the appropriate centroid block, MX[0:7] or MY[0:7], is fed to the most significant bits of the latch whilst **n**, NX[0:7] or NY[0:7], is connected to the least significant. This is shown on sheet Lookup 4. The address lines to the RAMs are LX[0:15] and LY[0:15] — the latter are passed to the Selector card as they conveniently offer a two clock delay over the MY/NY values. The arrival of an ECD signal from a point in its delay chain, 'ECDC', (see event validate) captures in the counters the **m-n** information for that event simultaneously in X and Y, using the 'LD LADR*' signal coupled to the counter parallel load inputs. The timing is shown in Fig. 28. The two latches and OR gate sampling ECDC on Lookup 4 eventually produce a two clock period active low pulse after the RAM address has loaded, 'LRAM EN*', which is used to enable the RAMs. Enabling the RAMs for events only reduces power consumption and the two clock period allows relaxed RAM timing (data rate 5MHz max). The RAMs are shown on Lookup 5 — after an access period the sub-pixel information is available as 'XSP[0:2]' and 'YSP[0:2]' and passed to the Selector card where there is sufficient time to pass through the data multiplexer on their way to the FIFO buffer.

The RAMs in effect perform a division and sorting of the centroid into bins, according to how they were programmed. The process of centroiding is illustrated in Fig. 29.

To load the lookup tables by software over the MACSbus, the pair of RAMs are treated as in parallel, that is, a common address is applied to them, and a 6 bit word transferred to or from them. This reduces table load time. Integrations cannot take place whilst the tables are loaded so a command issued to a MACSbus sub-address, 'OC LMODE*', puts the RAMs in a computer access mode where the 16 data lines from the MACSbus interface, via transparent tri-state latches, are connected to the counter/latches driving the RAM address lines, instead of the centroid blocks. This is shown on sheet Lookup 3. While in this mode, 'COMPACC*' prevents ECDs on the Darray card and disables the centroid MX/Y and NX/Y tri-state latches. A word to the next MACS sub-address loads a starting address, using the strobe 'OC LADR*', into the two counter sets. The transparent latches allow synchronous loading on PCLK to take place correctly with the asynchronous strobe signal. The timing of the operation is shown in Fig. 30. Then writes of data to a third MACS sub-address, using 'OC LDAT*', (an OR gate protects the tables against access when not in the correct access mode)cause the RAMs to be enabled, 'RAMWR*' gated to give 'LRAM EN*' on Lookup 4, and the coincidence with the RAM write signal causes words to be written into the lookup table. The rising edge of RAMWR* (or 'RD LDAT*') sets a latch whose output is sampled by PCLK. This generates (Lookup 4) a one clock period pulse of 'LA INC' which causes the address stored in the counters to auto-increment. This allows faster loading of the Lookup tables. When the tables have been fully loaded the computer access mode is disabled.

Reading back the contents of the lookup tables is a similar process as Fig. 31 shows. The read strobe 'RD LDAT*' is used instead, and it allows the contents of the RAMs onto the data bus MBD[0:15] via tri-state buffers. Again the auto-increment facility is available on the address.

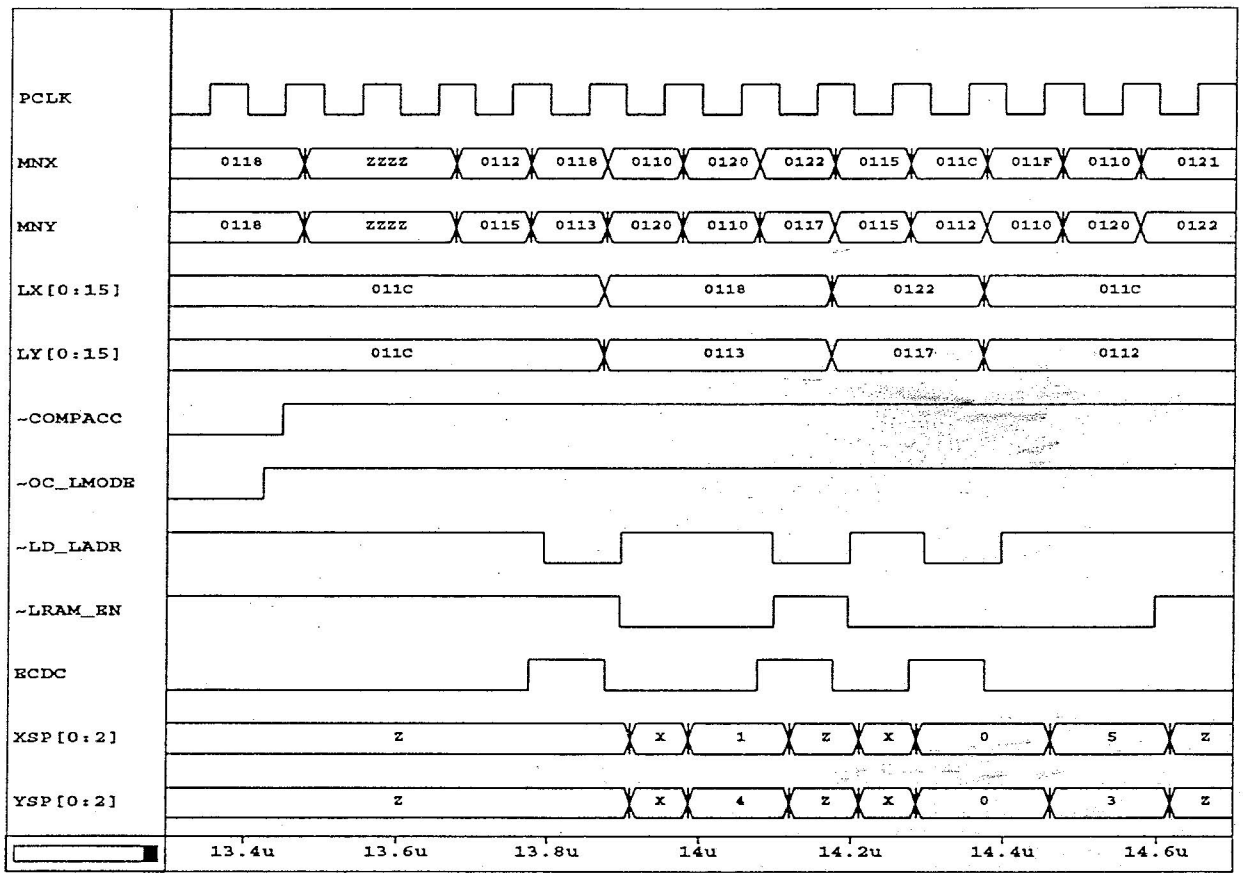

Figure 28: Timing for accessing the centroiding Lookup tables.

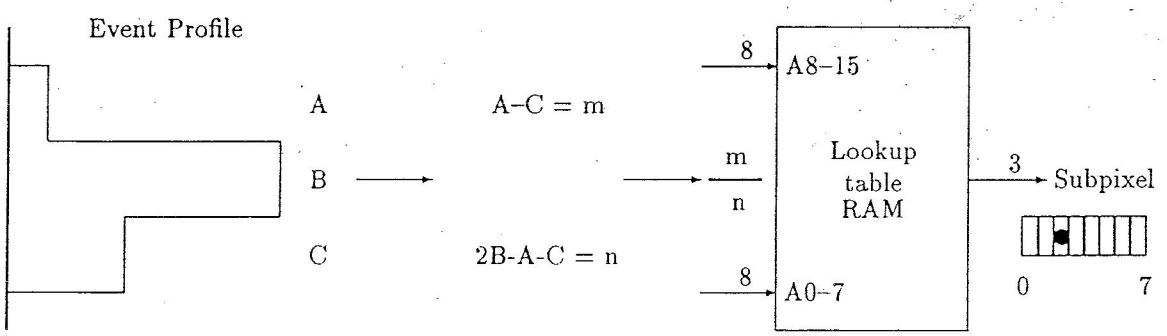

Figure 29: Event centroiding

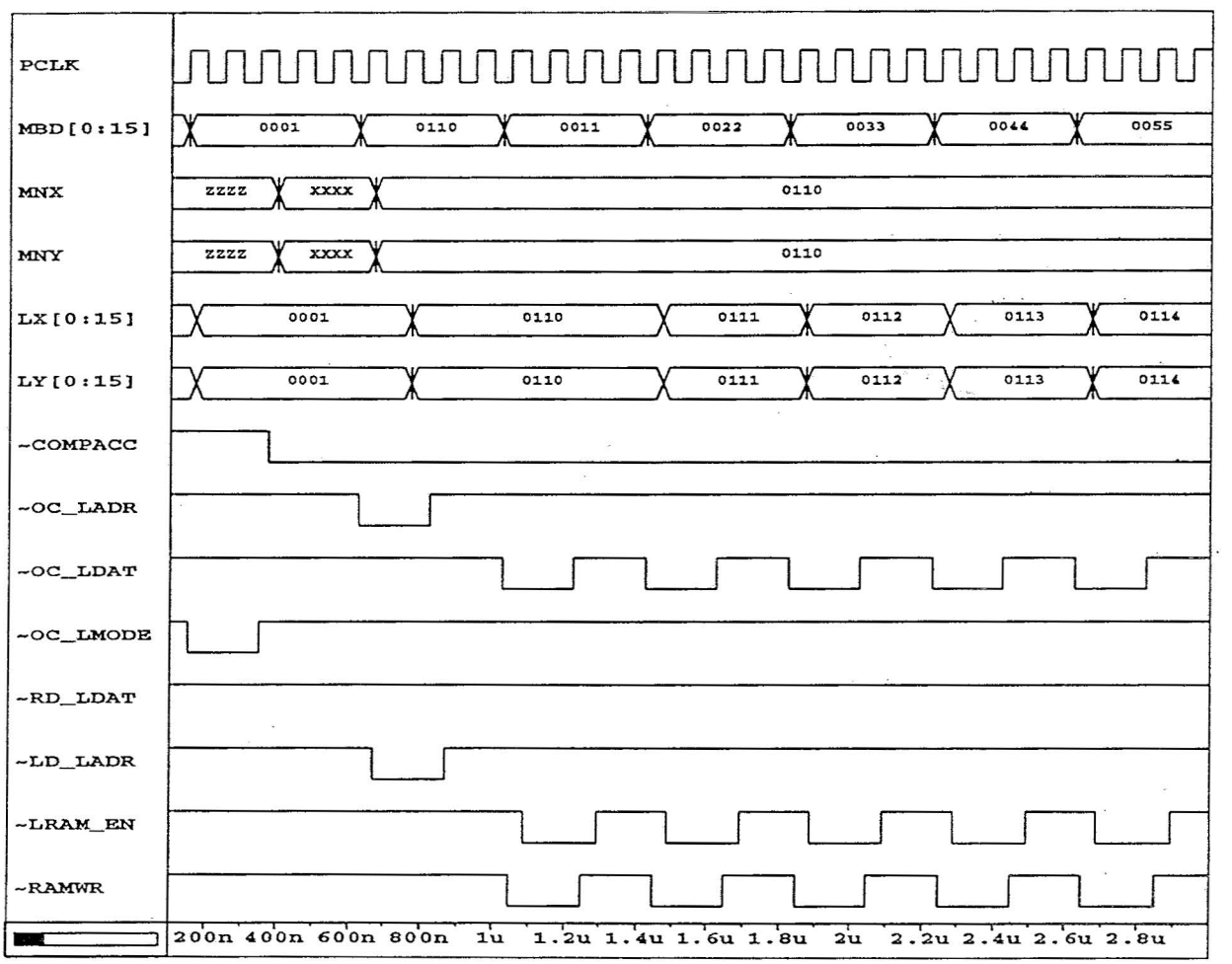

Figure 30: Loading the centroiding Lookup table RAMs.

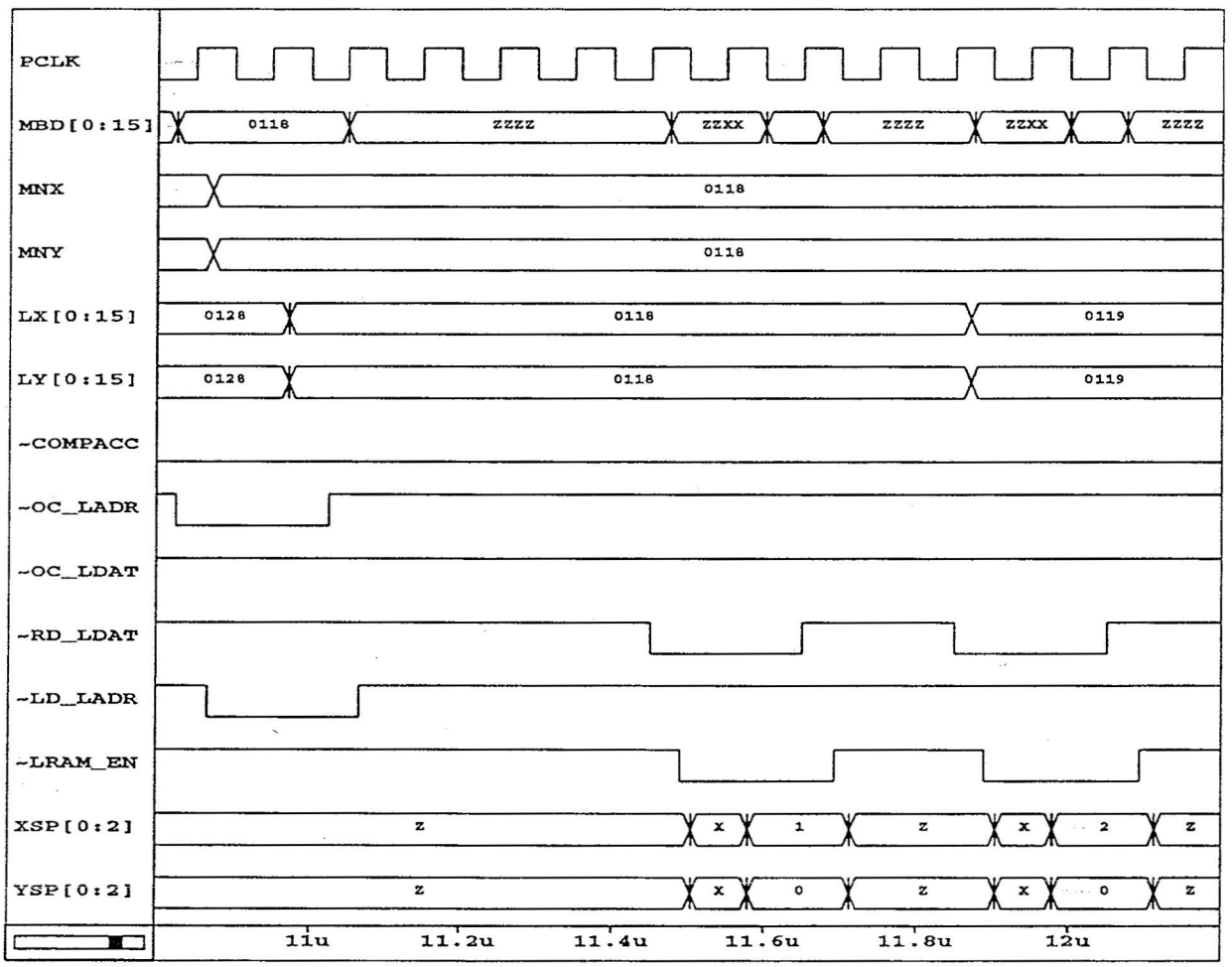

Figure 31: Readout of the centroiding Lookup table RAMs.

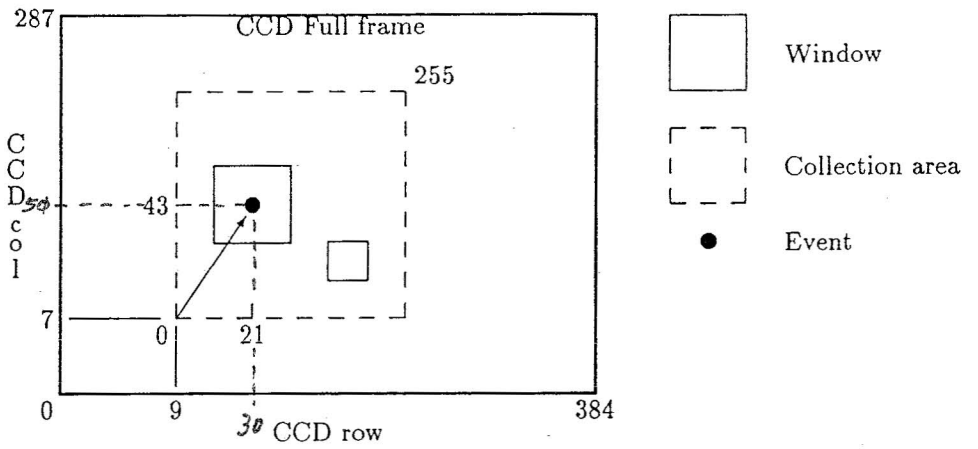

Figure 32: Collection area within a CCD frame

9 Data buffer and input selection: Selector card.

9.1 X/Y Counter offset

This block is the final source of information for formatting event positions by the FIFO multiplexer. Its purpose is to provide the values of X CCD pixel and Y CCD row for an event that are correct at the time its sub-pixel values are being presented at the multiplexer. The information source are the X and Y counters, XC[0:7] and YC[0:7], from the camera logic and offsets are added to account for the row and pixel propagation delays through the processing chain.

The link to the **DPU** imposes a constraint on the number of counter bits that can be used — this is 8 so only the least significant 8 bits of the X and Y position counters are used. An overflow 'wraps' around so any descrambling has to be dealt with in the **DPU**. Also, in the detector head, the size of the image intensifier with respect to the CCD gives a maximum area of 256 by 256 CCD pixels for the full image area, which gives in turn the maximum window size of 256 square pixels. The position of this data collection area is variable as it depends on the mechanical coupling of the intensifier and CCD, and the system optical axis. The **DPU** has a requirement that all CCD windows appear within this collection area and that the position of all events within this area to be referred to the origin of the collection area. This is illustrated in Fig. 32. For example, if an event was found at (30,50) in absolute CCD pixel coordinates and our collection area started at (9,7) in the CCD, the **DPU** receives an address of (21,43) referred to this frame. The restriction on the placement of windows is handled purely by software whereas the coordinate change is dealt with by extending the offset applied to the X and Y counters which operate in the CCD coordinate frame. The offset would be set by hardware links according to the detector head used.

This block is shown on sheet Selector 3. The Y offset is performed simply with an 8 bit adder giving a result YPIX[0:7]. The X offset requires an octal latch on the output of the adder in order that the result, XPIX[0:7], which changes at the PCLK rate, meets the setup time for the FIFO section (The Y address is set up well ahead of its first requirement by the arrival of the first event on a line). The offsets shown here on Selector 3 are, by default, chosen to position the collection area centrally on the CCD but will require adjustment.

9.2 FIFO Data selector

The purpose of this block is to multiplex all the sorts of information into a 23 bit word suitable for loading into a FIFO buffer. The type of information depends on the data acquisition mode in use, a normal data

mode or an engineering type. The sources of information include: The X CCD position (8 bits from the X offset block), the Y CCD position (8 bits), the X sub-pixel value (3 bits), the Y sub-pixel value (3 bits), the Double Count bit, the window ID (4 bits), the event height (8 bits), the event energy (8 bits), the X **m-n** values (16 bits) and the Y **m-n** values (16 bits).

There are 8 acquisition modes defined, so an 8 way multiplexer is used. The mode is selected using the output of a 3 bit latch (sheet Selector 3), giving 'MODEA1', 'MODEB' and 'MODEC'. The strobe 'OC IMODE*' loads this from the **MACS**bus (this can also be read back in the status word). The **DPU** link defines a 23 bit data word plus a parity bit. Encodings for the 23 bit field are shown in Fig. 33. It shows that a small saving can be made in making the Window ID field with a 4 way multiplexer only. The multiplexers are held on sheets Selector 4–9. All the information for a single mode must be available by the end of the same clock period. This requires delays on some of the fields — the Window IDs being the prime example (Delay block) as they are generated several clock cycles before the centroiding completes processing on the accompanying pixel. The first stage of the delay block (delay 2 cycles) is contained on the Camera card with the second stage, an additional 5 cycles, contained in Selector 4. The provisional window IDs, 'PID[0:3]', are input to a pair of counter/latches plus a simple D latch. With their load inputs connected to the inverted least significant X counter bit, XC0, each delays by 2 cycles. The final result, 'ID[0:3]', is connected to the 4 to 1 multiplexer on Selector 4, along with other information as indicated on Fig. 33.

However, the information for different modes do have varying arrival times. A final 1 bit wide 8 way multiplexer, on sheet Selector 10, selects a ECD off the timing chain (ECDA–E) to pass into the FIFO write logic, as 'SELECD*', so that the mode information will be written at the correct time. Let's use an example from Fig. 34 which is in mode 0 — the word written to the FIFO ('FIFOIN') is 3A6F85 hex. This is made up from the window ID (5) giving the least significant 4 bits. Then there is a 9 bit field for X. The XC/XSP value is 21 giving sub-pixel 1 — with mode 0 giving low resolution the 2 bit sub-pixel field, after right shifting, is 0. The offset is added to XC[0:7], giving BE hex. The least significant 7 bits are inserted in the 9 bit X field. Now for the 9 bit Y field we have YC/YSP equal to 26 from which 3 is placed in the 2 bit sub-pixel field, and with the offset added giving YPIX[0:7] a value of F4 hex, we insert the 7 least significant bits into the 9 bit field. Finally the 23rd bit is a copy of the 'DOUBLE' bit — in this case zero. With the fields concatenated, we obtain the FIFO input word.

9.3 FIFO Write control logic

This block accepts the formatted 23 bit word from the data selector and controls the writing into the FIFO together with some tag information. The data word is first latched, using counter/latches, as shown on Selector 4–9, to meet the setup and hold times for the FIFOs which are 512 words deep. The maximum write rate is 5MHz. The latching of the word is under control of the selected ECD signal, SELECD*, from the selector block. The ECD then goes on to trigger a one clock cycle write to the FIFO subject to some conditions being met. The FIFO must not be full, 'FULL*' low, — if so then data is simply ignored. The integration in progress latch must be set, 'INT ACT' — this changes state only at the end of FSYNC* (frame transfer) but is under software control via the **MACS**bus so data acquisition can be enabled as required. A latch is preset on the next PCLK and a one cycle pulse, 'W FIFO*', is coupled to the FIFO write enables (since ECDs are present for one clock cycle). The latches guarantee data is stable for the full write pulse. An example of timing using mode 0 is shown in Fig. 34 and this includes an example of two events at the minimum separation of 2 clock cycles. The effect of changing modes (this is not normally done during an integration) is shown in Fig. 35 for modes 2, 3 and 1.

A convenience feature is added to the write logic — 'pair logic'. In the engineering modes (modes 4-7 in Fig. 33) there are two items of information per event (eg. height and energy) and this feature writes first one into the FIFO then the second 2 clocks later. It saves integrating twice for the two data sets — fortunately the logic is simple and is shown on Selector 10. Setting MODEC high selects this logic via the gating which feeds a D flip-flop providing the 'MODEA' signal for the multiplexers. If MODEC is not high, then MODEA mirrors MODEA1. Otherwise, MODEA is the inverse of the signal 'LMODEA' which originates on Selector 9. Here MODEA is sampled by the FIFO input counter/latches so that LMODEA changes whenever an event is captured. This produces the toggling action on event arrival as shown in the timing diagram of Fig. 36. This shows first mode 4/5 then mode 6/7. We make use of the requirement that the second information word is available two cycles after the first (selecting an

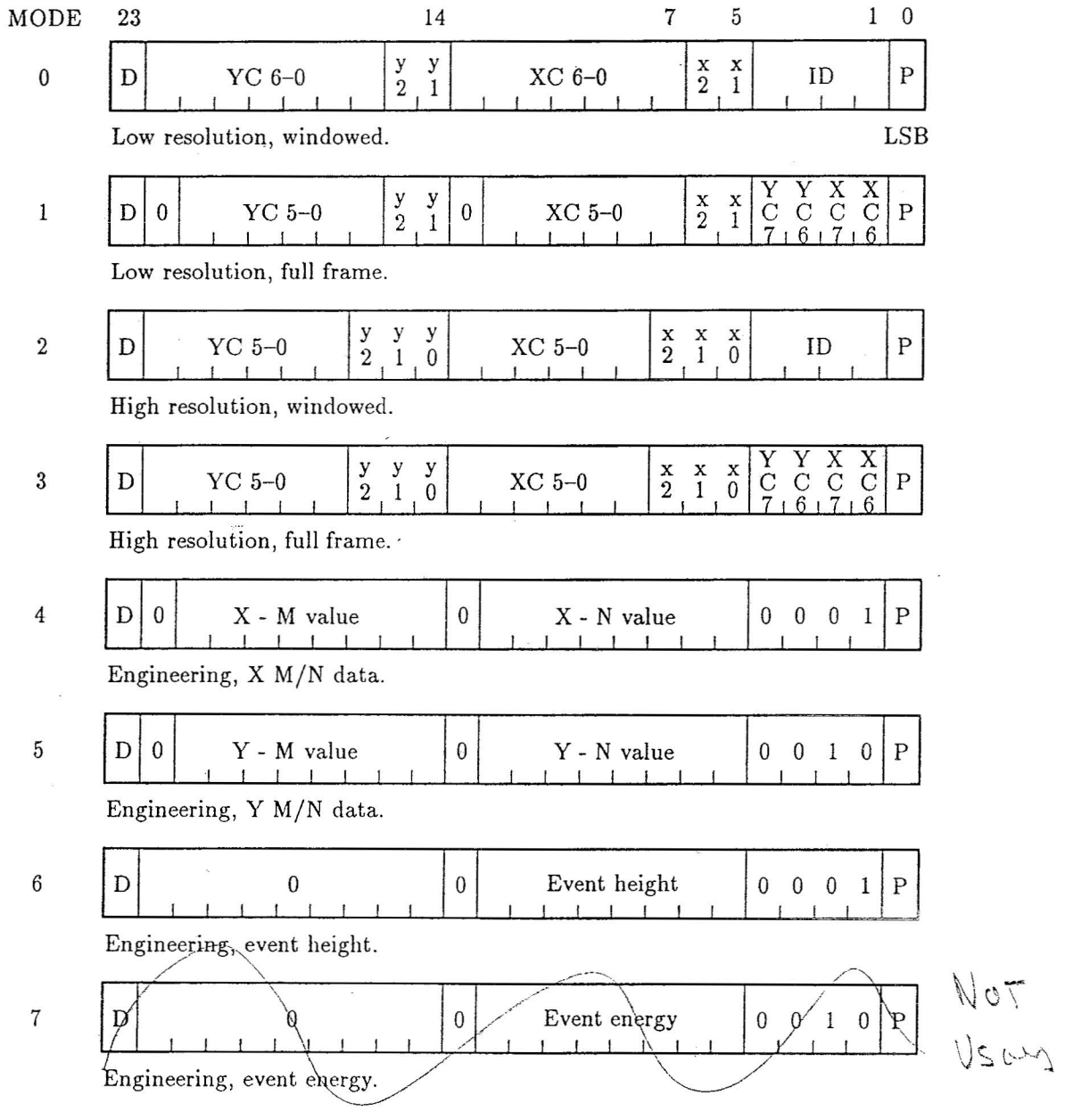

Figure 33: Provisional DPU data transmission formats

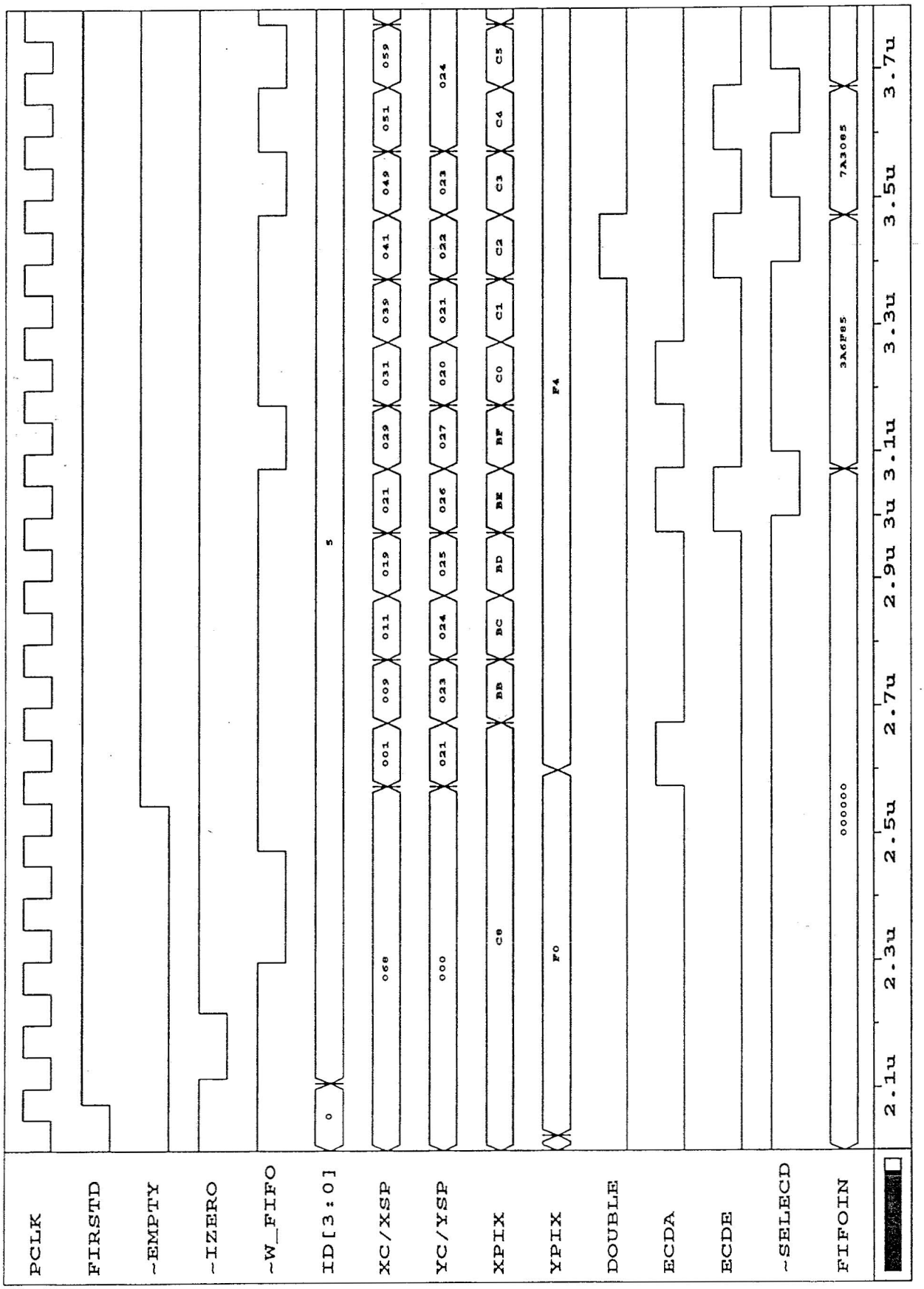

Figure 34: FIFO input timing for acquisition mode 0.

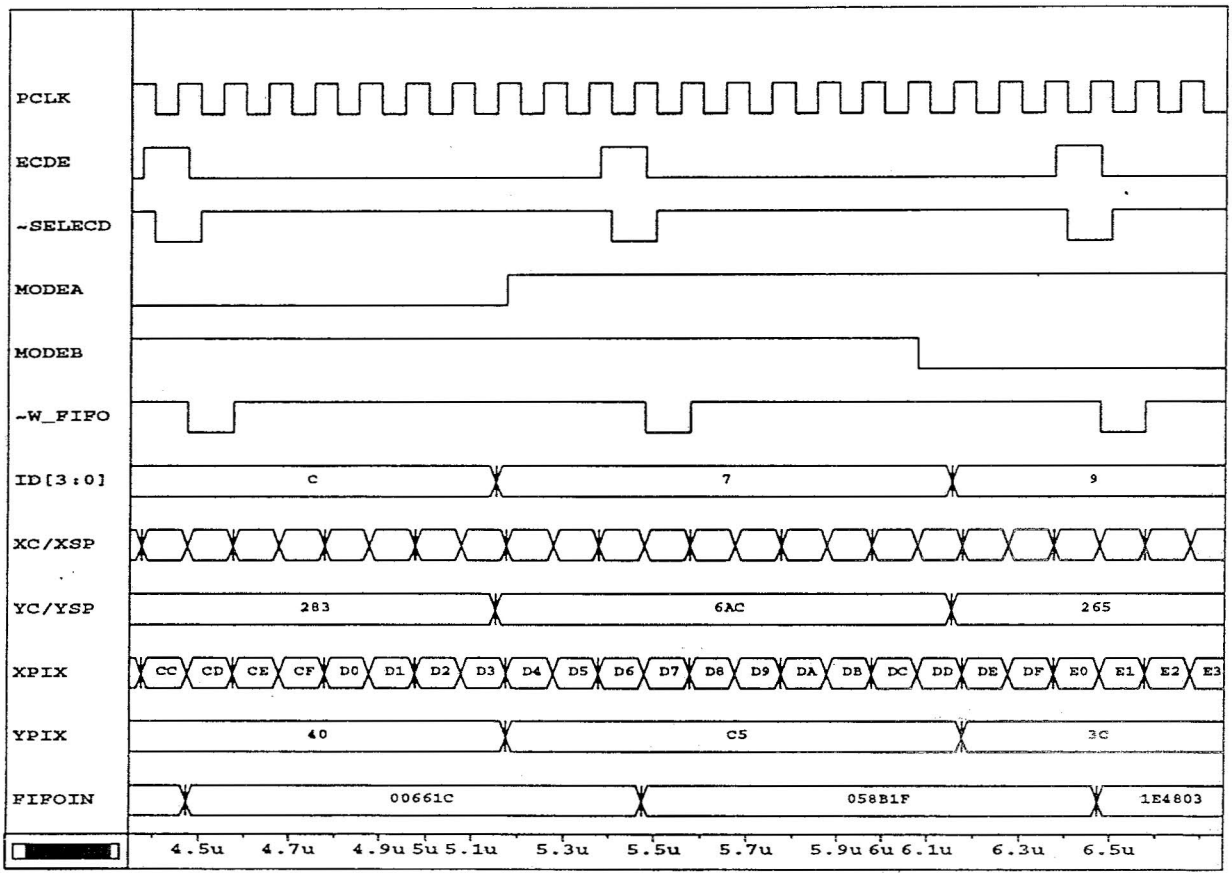

Figure 35: FIFO input timing for acquisition modes 1–3.

ECD to match the second). Additional words arriving in the first mode in the period is ignored. For the centroiding information we can use M/NX in the first instance, but need LY[0:15] in the second from the Lookup card as this is M/NY delayed by two cycles. There is a requirement that for the 'pair logic' to cycle in the correct order, the previous mode selection should have MODEA1 set low.

The write logic has an additional section, tied in with reseting the FIFO, to generate a frame tag. From the MIC ground based program it has been found that this form of FIFO can have problems. Very occasionally, FIFOs connected in parallel to give a wider word will get 'out of sync' with each other so scrambling the data. A reset of the FIFO each frame prevents any such problem, should it occur, from causing a long term problem. As the FIFO data sheet explains, the FIFO flags don't set as one would expect under all conditions. A little logic issues a FIFO reset pulse after the frame transfer phase is complete, at the start of the first non-dump data row to be read out of the CCD. The signal 'FIRSTD' goes high at this point and is sampled with a D type latch on Selector 10. A NAND gate generates a low going pulse at the rising edge of FIRSTD which, after combining with Local reset, goes on to form 'IZERO*' which is used to reset the FIFOs and the counter/latches on their inputs. This gives the maximum time to empty the FIFO — if, on average, the FIFO cannot be emptied by this time then the FIFO will spend much of the frame readout period in the full state, ignoring input data anyway, thus resulting in data loss. The FIFO reset itself can be removed if the FIFOs prove reliable.

The FIRSTD rising edge pulse is gated with an enable signal 'FTAGEN*' and sampled by a 2 clock cycle delay chain whose output presets the FIFO write pulse latch. This will be cleared on the next PCLK after the delayed pulse is removed, so forcing a word, after recovery from reset, of all zeros into the FIFO as a frame tag. The timing for this is shown on Fig. 34. An enable bit in the mode select latch enables or disables this extra write as, for example, it is not desired in full frame acquisition mode. An OR gate on Selector 3 ensures that these writes can only be done if an integration is active while forming the 'FTAGEN*' signal.

9.4 Integration startup

In order for data to be written into the FIFO buffers, an 'integration' must be started. Enabling and disabling writes allows a clean change between operating modes as the FIFOs will be clean and empty of the previous mode's data. The strobe 'OC INT*' latches data bit MBD0 into a latch to create the signal 'INTQ' (Selector 3). On power up, this latch is cleared. Enabling and disabling should take place at the end of the CCD frame transfer period when there will be no pending events for the FIFO, so INTQ is sampled at the rising edge of IRUN* to create the enable signal 'INT ACT'. The integration mode should be set before this. Timing for this point is shown in Fig. 37 (this also shows a status read using 'STATUS*').

9.5 FIFO Output and DPU link

This block reads out data words, when available, from the FIFO buffer and transmits them in serial form, with parity, to the **DPU**. Unfortunately the FIFOs data sheet highlights a problem with the FIFO flags which means that FIFO reads should not be truely asynchronous with respect to FIFO writes, so the FIFO read clocks are produced from the 10MHz system clock as are the writes. A read pulse must complete before, or at the same time, as a write pulse occurs. The circuitry is shown on sheet Selector 11. The FIFO empty flag, 'EMPTY*', is sampled by the serial link clock, 'LCLOCK', a buffered version, 'DCLOCK', of which, is passed on to the serial link buffers on the motherboard. If the FIFOs are not empty and a serial transmission is not under way, a serial transmission sequence is initiated. The timing for this is shown in Fig. 38. The synchronised flag, 'LEMPTY', acts as a transmission busy signal. With the aid of a D latch and gate the falling edge generates a one serial clock pulse 'RCYCLE' which is sampled, by PCLK on Selector 10, to produce a FIFO read pulse 'R FIFO*'.

By the rising edge of this the next data word is available on the FIFO outputs, OFIFO[1:23] (this is 3A6F85 hex in the example). The read pulse is connected to the shift/load input of a 24 bit shift register so this data is parallel loaded into the register ready to be shifted out. The read pulse delayed by one PCLK presets the previous latch. One serial clock after LEMPTY and 'BSEL' is driven low — this selects with the 2 to 1 multiplexer the shift register output, 'SDATA', to be coupled to the serial output

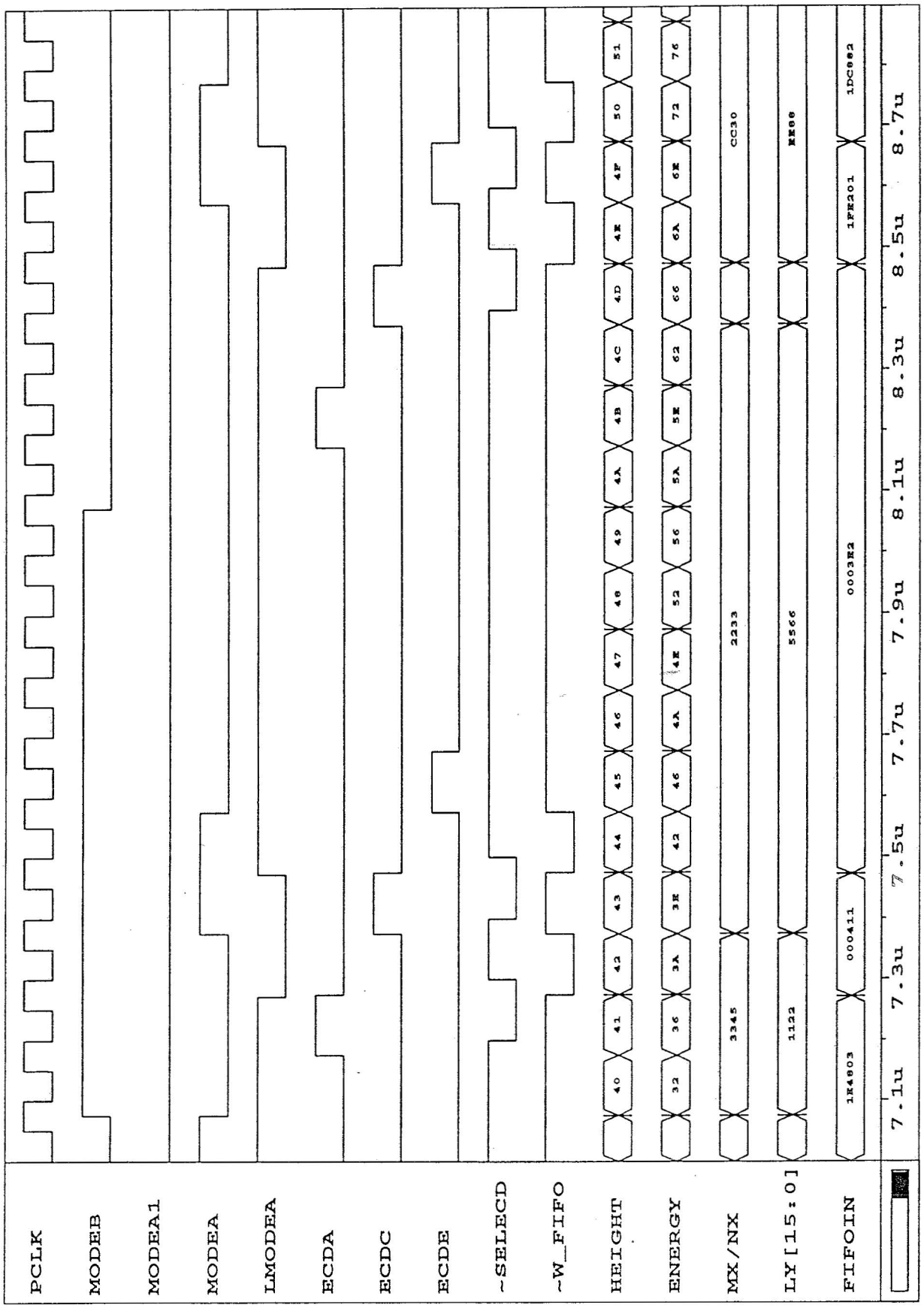

Figure 36: Timing of 'pair-logic' in data acquisition.

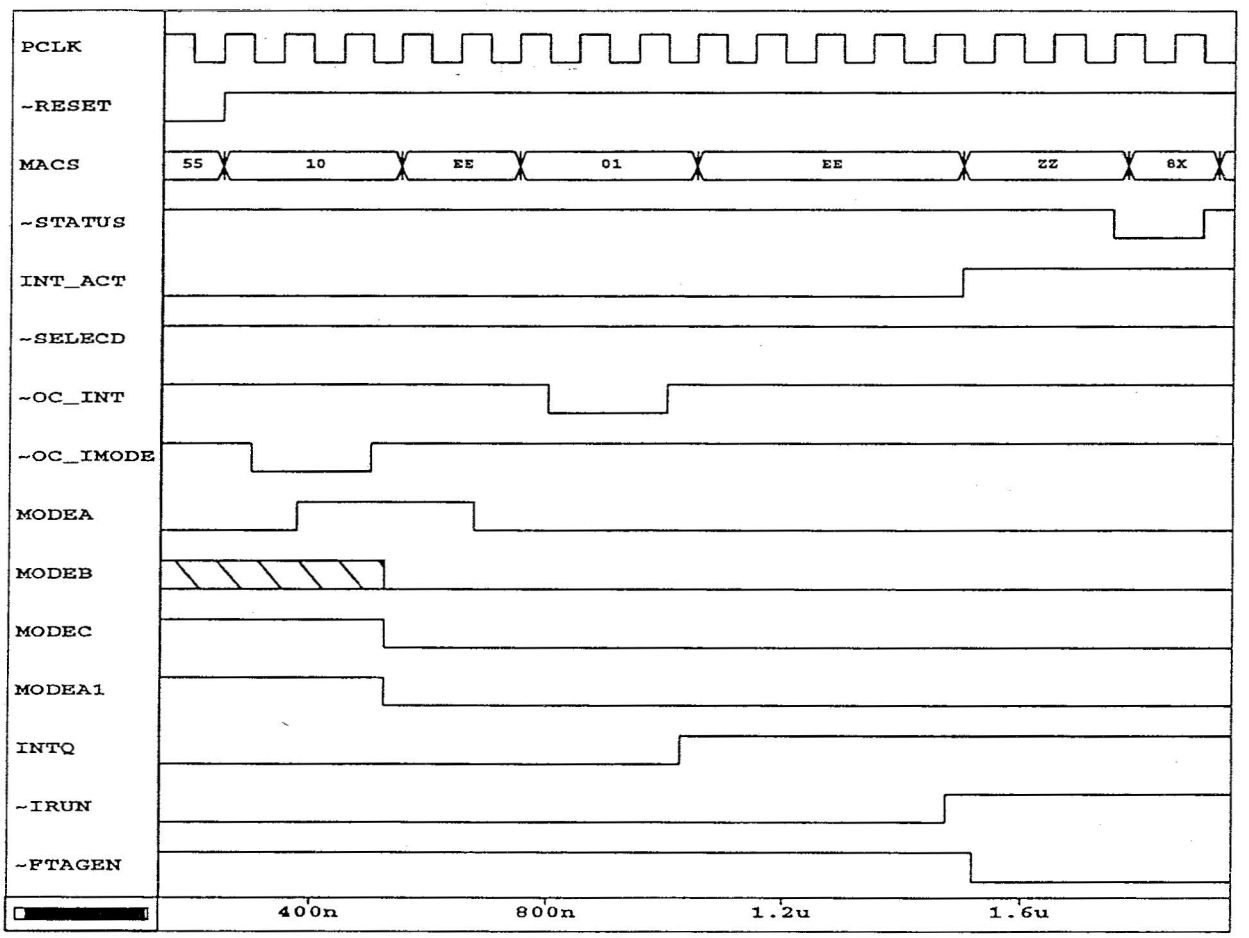

Figure 37: Timing for starting an integration with the FIFO buffer.

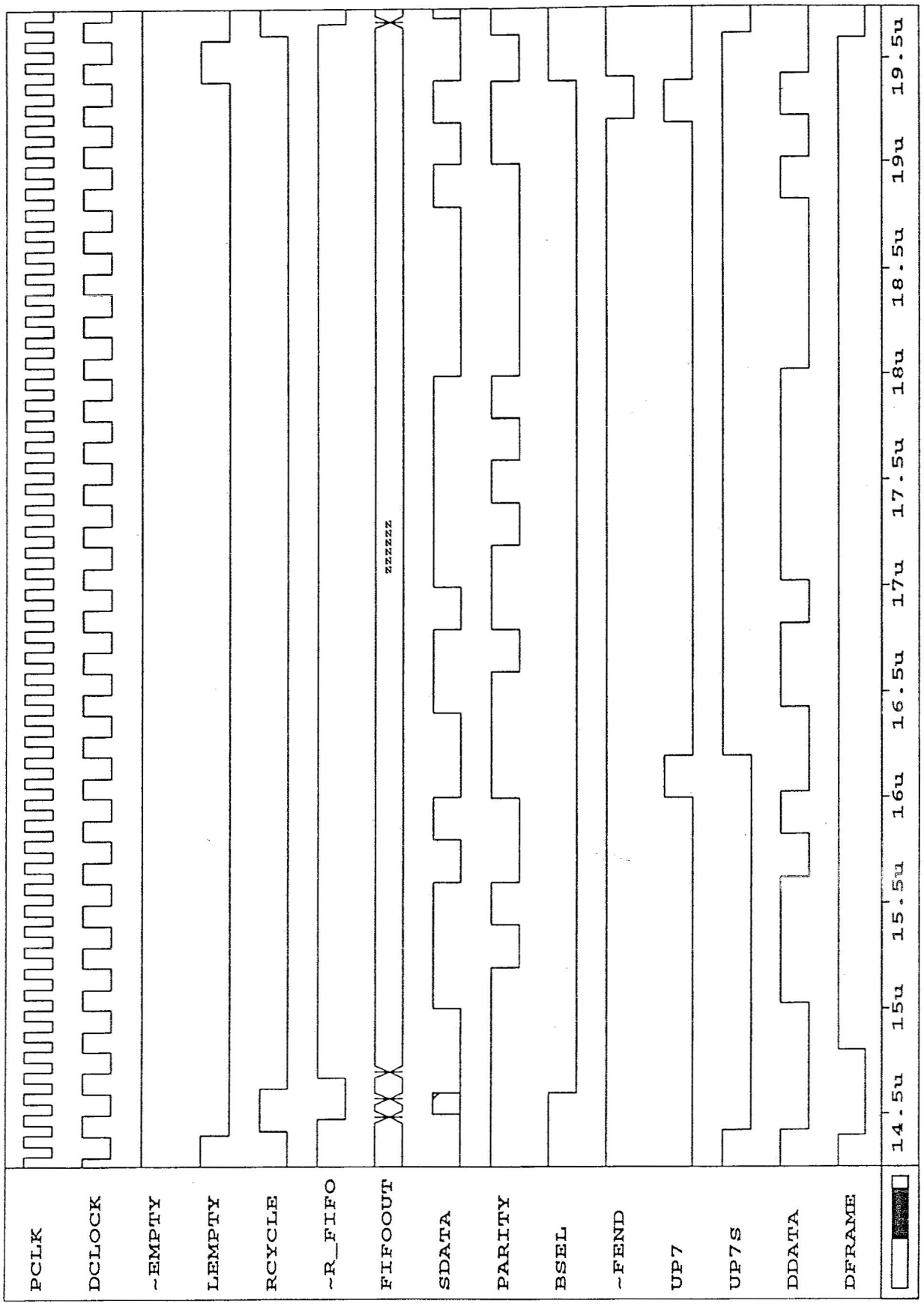

Figure 38: DPU link Data Transmission timing.

line 'DDATA' (to motherboard). Another LCLOCK later and the frame sync signal 'DFRAME' goes high. This is used to indicate a valid transmission on the serial link and at the same time it enables the DDATA output, formerly held low, and releases the inhibit input of the 24 bit shift register. The serial link consists of 3 signals, serial clock, serial data and the 'Frame' signal. At this rising edge of serial clock the first data bit (MSB) is available on the serial data output. On successive rising edges of the serial clock 23 bits are shifted onto the serial data line, with the output passing also into a serial parity generator, constructed with a toggling J-K flip-flop, which produces the 'PARITY' signal. Finally, for the last, 24th bit of the transmission, the output of the parity generator is inserted into the stream, to give odd parity including the parity bit (LSB). The DFRAME signal is then deasserted. A 4 bit counter plus D latch and OR gate counts the 24 bits. The counter load input is released with DFRAME high so 'UP7' goes high for the 7th bit and the latch ('UP7S') is high from the 8th bit onwards. The 2nd UP7 occurs during the 23rd bit and, combined with UP7S, gives 'FEND*'. This is returned to the input of the LEMPTY latch so that at the end of the 23rd bit LEMPTY goes high *and* BSEL goes high to select the PARITY bit. A LCLOCK cycle later this change reaches DFRAME. This implies a transmission time of 24 serial clock periods with a two clock minimum gap between transmissions. The most convenient serial clock rate is produced from the system clock so giving a maximum frequency of 5MHz. There are two serial link output sets provided, with independent differential line drivers — the buffers being located on the Mother board.

10 Mother Board: Mother card.

The prime function of this card is to interconnect the 6 cards of the processing electronics with connectors to the Detector head, EHT supply module, Power Supply, Cabling Harnesses and miscellaneous monitoring devices.

There are a few components on the board — buffers for the **DPU** link (differential drivers), on Mother 2, and for the System clock and detector interface (SOS devices). The system clock distribution, on sheet Mother 6, is performed in a single buffer device, the received master clock from the detector head being fanned out to each card on the backplane to minimise clock skew. The control signals for the detector interface are also buffered here.

The motherboard also distributes power and holds post-regulation components for the logic supply.

At this date, a full description awaits the design of the 'Monitor' card and decisions on the pin out scheme of the remaining connectors.

A Devices used in design

The following is a list of the SOS and bipolar devices used in this initial design with the approximate number used:

MA7001 512x9 FIFO	9
MA9187 64Kx1 RAM	10
54HSC00	4
54HSC02	4
54HSC04	3
54HSC08	8
54HSC10	1
54HSC14	1
54HSC21	3
54HSC32	8
54HSC74	16
54HSC86	1
54HSC109	8
54HSC138	3
54HSC151	19

54HSC157	10
54HSC163	39
54HSC164	3
54HSC165	3
54HSC244	11
54HSC253	2
54HSC283	39
54HSC373	7
54HSC374	49
54HSC521	3
54HSC540	8
54HST245	2
75ALS194	2
LM119	2
2N2907A	2
2N2222	2
FU02 M/C	1

B Board area required

The following table 8 is a summary of the different sizes of ICs used in this design. DIL packages are assumed. A multiplier is shown giving the equivalent size in 16pin DIL packages. The total figure can be

package size	No. of	multiplier	16 pin equiv.
14 (0.3in)	54	1	54
16 (0.3in)	127	1	127
20 (0.3in)	78	1.25	97
24 (0.6in)	10	3	30
28 (0.6in)	9	3.5	32
68 (LCC)	1	3	3
Total			343

Table 8: Package sizes used in design

used to estimate the board area. No connectors are included — their number depends on the quantity of cards employed. Note that the total does not include spares to cope with design error corrections.

C Hardwired setup parameters:

The design contains some setup parameters which are hardwired and will require some readjustment as the design proceeds through its life. These are;

1. The line start offset (Z), currently −20 as set by pullup/down resistors in the Camera board on AX[1:8]. This is set so that on D0R[0:7] when the first real pixel of CCD data appears, the X counter should be zero.
2. The fixed comparison values for the comparators giving the signals 'LEND*' and 'LINEND*' on Camera 6 must be adjusted by the same amount that 'Z' is adjusted to mark the end of the active pixels and of the horizontal readout.
3. The MACS address of the Processing electronics should be set by the links on Camera 14.
4. The X and Y offsets of the origin of the data collection area on the CCD may need adjustment according to the physical alignment of the CCD and image intensifier. These are on Selector 3.

5. Selection of the MACS bus clock rate (500 or 250kHz) by links on Camera 13.
6. Selection of MACS bus terminator resistors (required if module is the last unit on the bus) on Camera 9–12.

D Signal name glossary

This section lists the major signal names used throughout the cards of the system (not Mother), and shows the numbers of the sheets on which they are found. Eg. S2 is the sheet, Selector 2. A number in bold indicates that sheet is a source of the signal.

ACTIVE Signal set when MACS bus is active and a clock is being modulated. C10, C12, C14 and **C13**.

AX(1:8) Value to load X counter with — could be fixed value or computer loaded value. **C3** and C6.

AY(1:8) Value to load Y counter with — could be fixed value or computer loaded value. **C3** and C7.

BACC* Signal low when camera bitmap being accessed. C4 and **C5**.

BMD(0:3) Output of Camera bitmap (source of Window IDs). **C5**, C6 and C7.

BRESET* Buffered RESET*. **C3**, C4, C13, **S2**, S3, S4, S10 and S11.

CHAN-A High when MACS channel A selected. C10, C12 and **C13**.

CHAN-B High when MACS channel B selected. C10, C12 and **C14**.

COLB+C(0:7) Indicates total energy in columns B and C of an event. Adder carries are associated with it. **P10** and P11.

COLSUM(0:7) Total energy within a column of the event. **P10** and P11.

COMPACC* Signal low when computer can access centroid lookup tables. C2, D2, D8, P2, P6, L2 and **L3**.

D0R(0:7) Analysis array output being the current row of the CCD. D2, **D6**, D8, P2, P3 and P4.

D0-2RC(0:7) Represents the difference between the outer rows of an event. **P4**, P5 and P6.

D0+2R(0:7) Reperesents the sum of the outer rows of an event. **P3**, P5 and P10.

D1R(0:7) Analysis array output being the central row for an event. D2, **D6**, D7, P2, P3 and P8.

D1RA(0:7) Event central row delayed one cycle from D1R[0:7]. **P3** and P7.

D1RC(0:7) Event central row delayed 3 clock cycles from D1R[0:7]. P2, **P3**, P8, S2, S5 and S6.

D2R(0:7) Analysis array output being the deepest row, that read out of the CCD first. D2, **D6**, D8, P2, P3 and P4.

DAA(0:7) CCD video data, post black level subtraction, before analysis array. D3, D4, D5 and **D10**.

DARK(0:9) Reperesents the sum of the black level reference pixels per line. **D9** and D10.

DCLOCK Serial clock for DPU link. S2 and **S4**.

DDATA Serial data line for DPU link. S2 and **S11**.

DFRAME Frame sync' pulse for DPU link. S2 and **S11**.

DOUB Flag high if Double event found (before enable gating). P8 and P11.

DOUBLE Flag high for an event if it is Double and the feature is enabled. P2, **P11**, S2 and S9.

DOUBLEEN Status line high if Double count feature enabled. C2, C3, P2 and **P11**.

DUMP* Signal high only for non-dump rows being readout the CCD horizontal register. C2, C6, **C7**, D2, D3, D4, D5 and D6.

E(2:9) Represents the energy within an event (MS bits). P2, **P11**, S2, S5 and S6.

ECDA Signal high for one clock period when an event is detected. D2, **D8**, S2 and S10.

ECDC Signal ECDA delayed two cycles. D2, **D8**, L2, L3, L4, S2 and S10.

ECDE Signal ECDA delayed 4 cycles. D2, **D8**, S2 and S10.

EMPTY* Signal low when the FIFO buffer is empty. **S10** and S11.

EOVF Signal high when event energy exceeds 10 bits. P8 and **P11**.

ETHRES Signal high when event energy exceeds the threshold set. P8 and **P11**.

FIRSTD Signal rising edge indicates this is to be the very first real data row from the CCD. D2, **D6**, S2 and S10.

FTAGEN* Signal low when frame tag insertion into the **DPU** data stream is enabled. **S3** and S10.

FULL* Signal low when the FIFO buffer is full (512 words). **S10**.

HRUN* Control signal to operate the CCD horizontal clocks when low. C2, C6, C7, D2 and D6.

IBLK* Active low pulse to signal black level reference pixels arriving. C2, **C4**, D2 and D9.

ID(0:3) Window ID number timed to match up with its events. **S4**.

INT ACT* Low when an integration is active — synchronised to frame start. **S3** and S10.

IRUN* Control signal low to run the CCD image area clocks and also indicate the frame transfer phase. C2, C5, **C7**, D2, S2 and S3.

IZERO* Active low to reset the FIFO buffer and prepare a frame tag. S4, S5, S6, S7, S8, S9 and **S10**.

LCLOCK Base clock frequency for the **DPU** serial link. **S4** and S11.

LD LADR* Signal low to latch an address for the Lookup tables. **L3** and L4.

LINEND* Signal low for one cycle when last pixel in a row is being counted. C6.

LRAM EN* Signal low to enable the Lookup table RAM for access (computer or by centroiding). **L4** and L5.

LRESET* Signal low on Camera card to indicate the CCD clocks are stopped and computer access to the bitmap is possible. **C4**, C5, C6 and C7.

LSEND* Signal low for one cycle when last light sensitive pixel in a row is being counted. C6.

LSYNC* Signal high during a horizontal readout of data to indicate the light sensitive pixels. C2, C5, **C6** and D2.

LX(0:15) Represents the X centroiding information as captured to address the centroiding Lookup table, when an event is detected. **L4** and L5.

LY(0:15) Represents the Y centroiding information as captured to address the centroiding Lookup table, when an event is detected. L2, **L4**, L5, S2, S5, S6, S7 and S8.

M-N EN* Low when the centroiding information buffers are to be enabled (not computer mode). **P6** and P9.

MA(0:4) MACS sub-address bus for decoding. C3 and **C14**.

MBD(0:15) . Data bus for information from the MACS bus. Globally available and bi-directional. C2, C4, C5, C14, D2, D7, P2, P11, L2, L3, L5, S2 and S3.

MCLKA0 MACS bus differential clock line (channel A). C2 and C7.

MCLKA1 MACS bus differential clock line. C2 and C7.

MCLKB0 MACS bus differential clock line (channel B). C2 and C11.

MCLKB1 MACS bus differential clock line. C2 and C11.

MCLKV Selected MACS bus clock to controller. C13 and C14.

MDATAA0 MACS differential data line (channel A). C2 and C10.

MDATAA1 MACS differential data line. C2 and C10.

MDATAB0 MACS bus differential data line (channel B). C2 and C11.

MDATAB1 MACS bus differential data line. C2 and C11.

MDATAV Selected MACS bus data to controller. C13 and C14.

MDOUT MACS bus data output from controller before buffers. C10, C12 and C14.

MODEA Selection line for event information multiplexer (LSB). S4, S5, S6, S7, S8, S9 and S10.

MODEA1 Selection line as set by computer, before pair logic converts it to MODEA. S10 and S3.

MODEB Selection line for event information multiplexer. S3, S5, S6, S7, S8 and S10.

MODEC Selection line for event information multiplexer (MSB). S3, S4, S5, S6, S7, S8, S9 and S10.

MX(0:7) X m centroiding information (for Lookup table). P2, P9, L2, L3, L4, S2, S7 and S8.

MY(0:7) Y m centroiding information (for Lookup table). P2, P6, L2, L3 and L4.

NX(0:7) X n centroiding information (for Lookup table). P2, P9, L2, L3, L4, S2, S5 and S6.

NY(0:7) Y n centroiding information (for Lookup table). P2, P6, L2, L3 and L4.

OC BADR* Write strobe for bitmap address. C3 and C4.

OC BDAT* Write strobe for bitmap data. C3 and C5.

OC CMODE* Write strobe for camera operating mode. C3 and C4.

OC ETHRES* Write strobe for event energy threshold. C2, C3, P2 and P11.

OC IMODE* Write strobe for data acquisition mode. C2, C3, S2 and S3.

OC INT* Write strobe to control integration operation. C2, C3, S2 and S3.

OC LADR* Write strobe for Lookup table address. C2, C3, L2 and L3.

OC LDAT* Write strobe to load Lookup table data. C2, C3, L2 and L3.

OC LMODE* Write strobe to control Lookup table access mode. C2, C3, L2 and L3.

OC P12–15* Spare Write strobes for monitor card. C2 and C3.

OC PTHRES* Write strobe to load event peak threshold. C2, C3, D2 and D7.

PARITY Shows calculation of Parity for DPU serial link. S11.

PCLK Master pixel clock frequency (10MHz) distributed to all cards.

PCLK2 Master pixel clock frequency, split to ease loading to Process card.

PID(0:3) Provisional Window ID information, partly delayed. C2, C5, S2 and S4.

PTHRES(0:7) Copy of event peak height threshold saved. D7.

R FIFO* Read pulse applied to FIFO event buffer. S10 and S11.

RAMWR* Write strobe applied to Lookup table RAMs. L3 and L4.

RCYCLE High when a read cycle from the FIFO buffer and DPU transmission are under way. S10 and S11.

RD BDAT* Read strobe for reading bitmap RAM data. C3 and C5.

RD LDAT* Read strobe for reading Lookup table data. C2, C3, L2, L4 and L5.

RD P5–7* Read strobes (spare) for monitor card. C2 and C3.

RESET* Power on Reset signal, distributed through system. C2, C3, P2, L2 and S2.

RSTRB* MACS Read strobe issued by slave controller to decoding block. C3 and C14.

SELECD* Selected ECD signal used to control FIFO buffer write operations. S4, S5, S6, S7, S8, S9 and S10.

SRUN* Camera control signal, low when CCD storage section clocks should run. C2, C4, C7, S2 and S4.

STATUS* Read strobe to gate general status information onto the internal data bus. C2, C3, S2 and S3.

SYNC Separated Sync' pulse from MACS bus clock. C13 and C14.

TIMEOUT Signal high when MACS bus times out, giving up on looking for a clock on the current channel. C13.

VCLE Signal high to increment Y pixel counter during a vertical transfer cycle. C7.

VD(0:8) Video data stream from the CCD, pre dark subtraction. D2, D8 and D9.

VDA(0:7) Raw CCD video data synchronised to local clock edge. D9 and D10.

VGATE Gate signal high when bitmap output is valid for a row. C2, C7, D2 and D6.

W FIFO* Write pulse to the FIFO buffer. S10.

WIN ACT* Signal low when there are active pixels, in a window, on the current data row. C2, C6, D2 and D8.

WSTRB* Strobe issued by MACS slave controller to decode block for a write operation. C3 and C14.

X-OVF* Signal low when the X centroiding information overflows 8 bits. P9.

XC(0:9) Horizontal pixel counter outputs. C2, C5, D2, D9, S2, S3 and S4.

XINC* Signal low when incrementing the bitmap address after a computer access cycle. C4 and C6.

XMD(0:8) X m centroiding information before auto-ranging. P7 and P9.

XND(0:7) X n centroiding information before auto-ranging. P8 and P9.

XPIX(0:7) X counter after collection area offset applied. S3, S4, S5, S6 and S9.

XSP(0:2) X sub-pixel number — the Lookup table output. L2, L5, S2 and S5.

Y-OVF* Signal low when the Y centroiding information overflows 8 bits. P6.

YC(0:9) Row counter outputs (Y position). C2, C5, S2 and S3.

YMD(0:8) Y **m** centroiding information before auto-ranging. **P6**.

YND(0:7) Y **n** centroiding information before auto-ranging. **P6**.

YPIX(0:7) Y counter after collection area offset applied. **S3**, S4, S7, S8 and S9.

YSP(0:2) Y sub-pixel number — the Lookup table output. L2, **L5**, S2, S5 and S7.

E Circuit Diagrams:

Circuit Diagrams of the cards, 'Mother', 'Camera', 'Darray', 'Process', 'Lookup' and 'Selector' follow: