Preprint LAPP-EXP-98.01
January 1998

A VME based CCD imaging system for the VIRGO interferometer control.

F. Bellachia, D. Boget, T. Carron, D. Castelazzi, G. Daguin, L. Derome, C. Drezen,
R. Flaminio, X. Grave, J.C. Lacotte, F. Marion, L. Massonnet,
R. Morand, B. Mours, V. Sannibale, D. Verkindt
LAPP, BP 110, F-74941 Annecy-le-Vieux Cedex, France

abstract

Large interferometric detectors, like VIRGO, are planned to detect gravitational waves, by observing the motions of the suspended mirrors of a Michelson interferometer with kilometric Fabry Perot cavity arms. Due to the large sensitivity needed, the set up requires a real time monitoring of its mirrors positions, and the analysis of beam shape during the alignment. In this paper, we present the development and the performance of a beam imaging system which uses a CCD camera interfaced to a VME board.

keywords:
Gravitational Waves detectors, VIRGO, imaging system, CCD cameras, VME board, positionning device.

Submitted to Nuclear Instruments and Methods in Physics Research Section A

1 Introduction.

The VIRGO interferometer [1] will have to detect spacial distorsions induced by gravitational waves, with the sensitivity $\frac{\Delta L}{L} \simeq 10^{-23}/\sqrt{Hz}$ around 100 Hz. To reach such sensitivity, the detector performances are pushed to the limits of the existing technology: The source is an ultra stabilized high power Nd YAG laser. By using light recycling techniques, Fabry-Perot cavities in the interferometer arms, very low losses mirrors, the detection can be optimized. To reduce the seismic noise contribution at low frequency, the mirrors are suspended to a multistage pendulum system.

To bring the interferometer towards its optimal working configuration, we must provide an accurate control for damping the large amplitude mirrors pendular motions, and monitoring of the laser beam shape. This can be achieved by using CCD cameras. A first imaging system has previously been developed and used mainly for beam observations [2].

In this paper, we detail the development of the upgraded version of such a system [3], not only for beam imaging, but also for mirrors motions control. We use a commercial CCD camera coupled to a VME interface. The development of this VME/VSB board is detailed. We have implemented different modules for the camera control, read out, and data storage. The board also provides external synchronisation facilities. We present a simulation of the camera data acquisition in memories and data read out on the VME bus. The electronic noise and the position resolution of the imaging system are presented. Finally, the imaging software tool is briefly discussed.

2 Imaging Applications.

One important request on our imaging system is its compatibility with the interferometer control system. This control is implemented in a distributed processing scheme. The real time processors are located in VME crates interconnected with real time data links and computer network. Unix workstations are used for the Graphical User Interface.

The aims of this imaging system are to monitor and store the mirror alignment parameters and to control the mirror positions by damping the large pendular motions during initial alignment. These low frequency oscillations have to be damped by a digital servo loop running at a few ten Hertz. Therefore, it requires that an image can be captured at a well defined time in response to a user supplied strobe pulse, to synchronise the different real time tasks. It also has to provide beam shape analysis and beam tracking during the alignment start up of the interferometer. These applications may need that the user could control the exposure time of the sensor for low level beam.

3 The camera.

The commercial asynchroneous EEV-CAM17-16/X219 camera [4] is used. The CCD is a frame transfer sensor which provides an image of 512x512 pixels in a non interlaced format, at the maximum rate of 60 frames per second. The pixels size is $15 \times 15 \mu m^2$ giving a square imaging area of $7.7 \times 7.7 mm^2$. The camera provides its own clocking sequences at 20 MHz frequency. It digitizes the analog signal (integrated charges) through an 8 bits ADC.

Compared to conventional video cameras, this model has many avantages: It has different operating modes like continuous or asynchroneous, it enables integration time and gain control.

In free running mode, image capture occurs continuously at the full image rate (60 Hz maximum). Asynchroneous mode allows the user to control both the exposure period and the time at which the exposure occurs. Integration is controled by the duration of a pulse and should be timed to coincide with the event to be captured. The exposure period may be of any duration (limited only by shift smear or saturation). An additional gain stage is implemented for low light operations.

The pixel clock (a key feature for our application), field synchronisation, and framestore 'GRAB'(which defines the valid pixels) signals are all provided with differentially driven outputs in order to simplify integration with framestores and image processing systems. The camera also provides a video signal, which is not compatible with European CCIR standard. The camera provides line and field period, respectively of $32\mu s$ and $16ms$. The CCIR standard requires $64\mu s$ and $20ms$. Therefore the video output can be displayed on a monitor, but is horizontally compressed.

4 The VME/VSB camera interface.

The VME interface was designed at LAPP in Annecy. It provides the camera power supply, the video with the CCIR format, the storage of the digitized image in memories, a DAC for gain control, a module to catch external trigger (or to built internal trigger) for the image capture, and a module to built the integration time signal with the desired length. Finally, we have implemented a VME interruption module to synchronise the data acquisition with external events or device.

4.1 The memories module.

The digitized data are stored at the rate of 20 MHz in four 128 Kb memories of 8 bits (Mem 0 to Mem3 on figure 1). Normally, 64 Kb memories should be enough to store 8 bits data, but we have extended the storage capacity of our interface to be able to handle camera models with 10 bits dynamic ADC. A multiplexor selects alternatively the different sets of memory ships, for writing operations. The two LSB (D0 and D1) of the write counter address are decoded to build the chip select memories signals (CS0, CS1, CS2, CS3), to allow the permutation between the four memories on each clocking signals of 50 ns period. Another multiplexor allows the decoding of the read/write address. During the first half period (25 ns) the data are written in memories. During the second half period, the four memories are opened simultaneously to set four pixels of 8 bits on the 32 bits VME or VSB bus. The memories can be randomly accessed (I0/I16) through the VME, by setting the start address of a memory zone of interest to be read out. The memories can also be filled through the VME drivers to test their fonctionnalities. The same data can be read out a few times for slow trigger rate.

4.2 The integration time module.

When asynchroneous operation has been selected, the camera goes into standby mode at power on. In this state any charges collected by the CCD is removed by the antiblooming structure, and the camera therefore does not appear to be sensitive. The camera is waiting for a signal to indicate when to capture an image.

As illustrated on figure 2, the rising edge of the external signal triggers the strobe signal STR which is held at a logic low and which defines the integration time (Ti). The STR signal is entirely built by a logic on the interface. It is generated using a counter started by the trigger pulse and incremented at each line. Therefore it is a multiple of the line period ($32\mu s$). The user sets its value in a register. At the end of the STR signal the charges are transfered (field transfer) in the non active area of the CCD in less than $260\mu s$. Then the charges are converted by the ADC and stored in memories. The camera produces the GRAB signal which is held at a high level from the beginning of charge transfer until the end of frame read out.

In fact a new integration can start immediately after the field transfer. The integration on the sensor before the end of acquisition doesn't affect the informations stored in memories until the beginning of field transfer on the sensor, so integration can be anticipated. It is then possible to make image acquisitions at the rate of about 60 Hz for integration times less than the ADC conversion time ($17ms$). For longer time, the integration starts immediately after field transfer, and extends after the falling edge of the GRAB signal. In this case the maximum trigger rate is function of the integration time and is less than 60 Hz. We build the VETOSTR signal to inhibit too early trigger pulses which would generate charges dumping at the wrong time and therefore loss of image. The VETOSTR signal is generated by the integration time module using the Ti value. It is adjusted to a high level in such a way that the integration period always finishes just after the GRAB falling.

We have also foreseen a default internal trigger acquisition signal at 50 Hz, so that the camera can work in asynchroneous mode without external device to trigger it. These external or internal trigger modes of operation are set by software, or by a switch in the front panel of the interface. The external trigger impulsions are provided to the interface through a BNC input in the front face.

Very short exposure periods result in a significant amount of frame shift smear, seen as vertical smearing of the image if the intensity of light on the sensor gets strong. We have fixed the minimum strobe pulse length around $100\mu s$. Strobe pulses of duration less then two lines periods will be ignored by the camera. At the same time, it reduces the risk of false triggering of the camera due to noise. For low light level, we have fixed the maximum integration time to one second (15 bits counter).

4.3 The video module.

The video produced by the camera can not be properly used for monitor display, because it does not fullfill the CCIR timing specification. On the VME interface, we have implemented a module able to produce the correct video with $64\mu s$ line period and 50 Hz field rate. The principle is illustrated on figure 3.

We have chosen a 256x256 resolution video image reconstruction, which is enough for monitor display. The raw digitized data produced at the rate of 20 MHz are selected by the read/write logic module. The strategy is to select one pixel out of two and apply this selection one line out of two. The 256x256 pixels are stored in a 64 Kb memory. The data are written in memory just during the time that the digitized data are produced by the camera. On the other hand, the memory is permanently read by clocking the data with a 5 MHz clock in the register of a video DAC. This device makes the conversion of the digitized information in analog signals mixed to the right synchronisation sequences, field and line blanking signals. The level of each kind of signals is automatically adjusted, so that we just have to provide TTL signals.

The video image refreshment time will depend on the camera operating mode. It depends directly of the trigger period. For slow rate, the same image can be read a few times before refreshment.

4.4 The interruption module.

This module has been designed to provide interruptions on VME or VSB bus to synchronise external events with the pixels acquisition in memories. The interruption can be generated at the end of any line acquisition. This line number is just set by software in a register. At each start of frame, a counter starts to count the number of lines, until it reaches the specified line number. An interruption is then simultaneously directed on a BNC output on the front face of the interface, and towards a VME/VSB logic ship. This module manages the request, taking into account the priority level. When the request is enabled, the address of the interruption routine to be executed is set on the bus to be directed towards the VME processor.

4.5 The hardware simulations.

For hardware implementation, some logical modules have been grouped in programmable logic devices. The MAX7000 family of high density cells and high speed CMOS-PLCC from Altera has been selected. These devices are supported by Altera MAX+PLUSII development system [5], a package that offers waveform design entry, compilation and logic synthesis, simulation and programming software. It provides VHDL, Verilog and other interfaces for additional design entry and simulation support from other standard tools.

Each fonctionnality (trigger+integration, interruption, video logic, acquisition logic, multiplexors, drivers) has been programmed and tested by the simulation. For instance we have checked the acquisition sequences of pixels in memories corresponding to the logical modules illustrated on figure 1. It includes: The acquisition logic module, the multiplexors module, the VME drivers and bus, the camera drivers, and the four memories. In this case, we simulate the data read/write in memories during camera acquisition. The timings are illustrated on figure 4.

The data acquisition in memories is driven by the 20 MHz pixel clock. During the first half period of 25 ns, the data coming from the camera through 'driver Cam' are set (gate OEDCA=0) on both (DBAI) and (DBAP) internal buses. For exemple, when two pixels 0x2B2C (fourth marker) are presented on these two buses, then Mem0 is only selected (CS0=0, CS1=CS2=CS3=1) for write operation (validation on WEMA rising edge). At this moment, 0x2C is writen in Mem0 at 0xB ADDMA address. At the next period (fifth marker), the data 0x2D2C is presented on the memories internal buses. The data 0x2D is stored in Mem1 (CS1=0, CS0=CS2=CS3=1) at 0xB address. At the next period again (sixth marker), 0x2E is stored in Mem2 (CS2=0, CS0=CS1=CS3=1) at 0xB address. Then (seventh marker), 0x2F is stored in Mem3 (CS3=0, CS0=CS1=CS2=1) at the same address. At the next period, we will write the data 0x30 in Mem0 at the address ADDMA=0xC and so on... The write address has to change only each four pixel clock periods, because we store successively the pixels at the same address but in the four different memories.

The data read out occures at any time i.e when the memories are not in write access. In this sequence, the read out occurs on the second half of the pixel clock period. This read out is quite slow, because of limited speed on the VME or VSB buses. The VME drivers are opened at the FR1 timing rate. These drivers are strobed on the rising edge of CRSA, when FR1 is held on a high logic condition. On our exemple, it corresponds to the positions of the three first markers. Data can be strobed a few times in the VME drivers during FR1 high level signal, so they must not change, which means that the read out address must not change too. For each read out, the four memories are opened simultaneously (chip select CS0=CS1=CS2=CS3=0) at the same read out address. For exemple (second marker), we read the two pixels 0x01, 0x00 on DBAP, and the two pixels 0x03, 0x02 on DBAI at the address ADDMA=0x0. Joined together, these four pixels form a 32 bits data on the VME bus (DB). The read out address just changes on the falling edge of FR1 signal.

5 The imaging system.

The imaging system is illustrated on figure 5.

The camera and the VME interface contain all the fonctionnalities described above. A video monitor can be connected to the video output of the interface for a direct image display. The software is based on the client/server model: A real time program called the image server, reads the image stored in the interface memories. The client, running on a workstation, displays the digitized images and proposes a wide range of tools for image analysis. The communication between the client and server programs is established by a specially developed software using TCP/IP protocol [6].

5.1 The performances.

The position resolution of the camera and its read out electronics has been evaluated. When doing a weighted average to compute the position of a beam, the resolution (σ_x, σ_y) is a function of intrinsic parameters such as the pixel size (Δ_x, Δ_y), the noise per pixel (σ_a), and the ADC dynamic range (d) [2]:

$$\sigma_i \simeq \frac{\sigma_a \Delta_i}{d} \tag{1}$$

The best resolution is obtained with a bright spot using the maximum dynamic of the sensor (large number of ADC counts) and is independant of the spot size. In order to calculate the position resolution of the system, we first evaluate the electronic noise. It has been measured by histogramming the 262144 ADC values of a 'dark' image after ambiant light subtraction. The result is shown on figure 6. The rms value (σ_a) is about 0.45 ADC count.

It has been compared to the fluctuation noise of a given pixel at a constant light level. The rms value obtained in this case, for 500 successives images, is still about 0.45 ADC count. The figure 7 illustrates the distribution.

For an 8 bits ADC and square pixels of $15\mu m$ x $15\mu m$, the theorical resolution is therefore $\sigma_x = \sigma_y \simeq 30nm$ per image. Since the camera produces a maximum of 60 images per second, the maximum position resolution can also be expressed as $\sigma_x = \sigma_y \simeq 4nm/\sqrt{Hz}$.

However, one should notice that the r.m.s noise increases when a gain is applied on the analog sensor output. For a maximum gain of 20 dB, we have measured $\sigma_a \simeq 4$.

To measure the camera position resolution, one creates a static spot on the sensing surface, by illuminating a pinhole rigidely mounted in front of the camera. The image is read and the spot position is computed at 20 Hz. The results are written in a file for each sample. For 100 measurements, figure 8 represents the histogramms of the positions variations on x and y axes.

The resolution is less than 50 nm for fast positions fluctuations. These results are in agreement with the expected value of 30 nm. A FFT calculation gives the spectral position noise along the x and y directions. The figure 9 shows these noise amplitudes. At high frequency, the measured resolution reaches the expected value. At low frequency, the resolution is reduced due to the slow thermal drift of the camera.

6 The imaging software.

The imaging program is designed on the client/server model. A real time program acts as 'image server' and is run by a VME based processor. It reads the image stored in the camera interface memories, and performs the calculations for real time applications. The client is a program running on a workstation to display the digitized images with various options choosen by the graphical user interface.

The communication between these two programs is established using 'Cm' [6], a TCP/IP based communication tool developed within the VIRGO collaboration. 'Cm' is designed in a way that one can easily write control-command applications in a distributed and heterogeneous environment, as for exemple communication links between clients and servers operating on different machines.

The programs are written in C language and the client uses Motif as Graphical User Interface.

6.1 The image server.

The image server program runs on a VME CPU. It reads the data stored in the interface memories (full image or restricted areas), and performs all the calculations for the alignment options. It can correct the raw image from distorsion and offset by applying relative gain or subtracting pedestal (for exemple ambiant light) directly after pixels read out. Calibrated images are then send to the client for display.

6.2 The Graphical User Interface.

The client has first to establish the connection with the image server, then the user can define the camera configuration (trigger mode, integration time, gain value, interruption mode...) and communicates it to the server to set correctly the hardware. The images are displayed in three dimensions or in matrix (x,y) representations with a color or grey level scale (from 0 to the maximum ADC dynamic). One image can be fully or partially displayed. In that case, the user can select with the mouse a box in the global image, and only this selected area will be processed and displayed. This means that the server will only read this area in the camera interface memories, and that the various calculations will be restricted to this area. This method is used to compute the positions of different spots on the mirror surface for local control applications [7]. It reduces the read out and the calculation times, which is very important for real time application. To speed up the image refreshing rate in the client, the pixels can be clustered together (2x2, 4x4, 8x8), giving a representation with reduced resolution (64x64, 128x128, 256x256). The user can also select a line (x) and a column (y) with the mouse, to build intensity profiles. Additional tools are available to caracterize the beam profiles like statistical calculations, gaussian beam decomposition, waist calculations... Specific menus have been developed for real time applications (local control of mirror position) and for the initial alignment of the interferometer. Image archiving is also available in this program. One typical view of the image display is presented on figure 10.

7 Conclusion.

We have designed and tested a new imaging system suitable for the VIRGO experiment needs. The image rate could go up to 60 Hz. The sensor size ($7.7 \times 7.7 mm^2$) is large enough to accomodate large beam displacements. We have measured a position resolution as low as ($4nm/\sqrt{Hz}$), in agreement with the expected value. The production of the VME interfaces is now in progress in order to start the installation on the VIRGO experiment.

References.

[1] C. Bradashia et al. - 'The VIRGO project: a wide band antenna for the detection of gravitationnal radiation' - Nucl. Instrum. Methods A 289, 518 (1990)

[2] A. Bazan et al. - 'A VME based imaging system for the VIRGO project' - Astroparticle Physics 2(1994) 229,234

[3] C. Drezen - 'Utilisation de caméras CCD pour le contrôle du détecteur d'ondes gravitationnelles VIRGO: Elaboration d'une interface VME et analyse d'images' - University of Grenoble thesis LAPP-T-97/07 sept(1997)

[4] EEV Imaging, Sensing and Displays Division - 'Super photon CCD camera CAM17-16/X219 user guide' - sept(1997)

[5] Altera Corporation - 'MAX+PLUS II Programmable logic development system and software' - v7.2 (1997)

[6] C. Arnault, P. Massart - 'Cm: A multitask communication package' - v7.3 sept(1996)

[7] H.Heitmann, C. Drezen - 'Measurement of position and orientation of optical elements in interferometric gravity wave detectors' - Rev. Sci. Instrum. 68, 3197 (1997)

Figure 1:

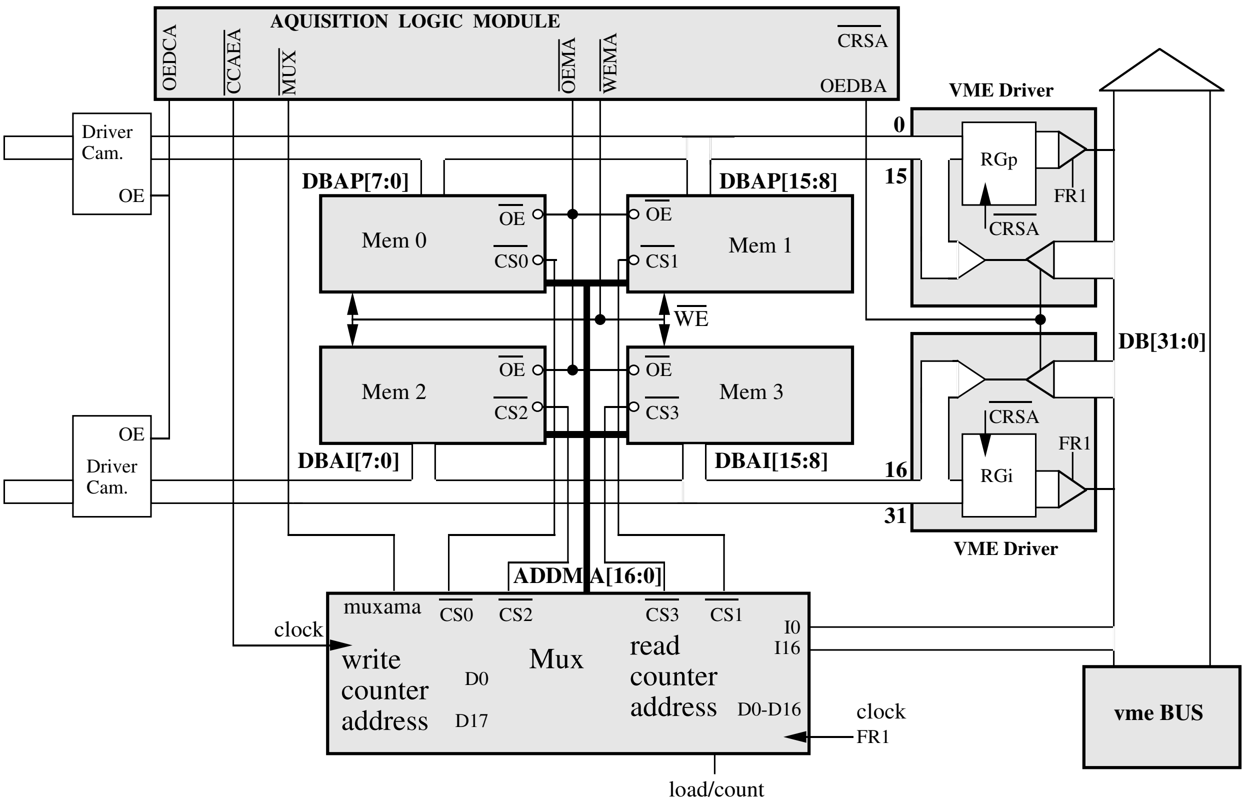
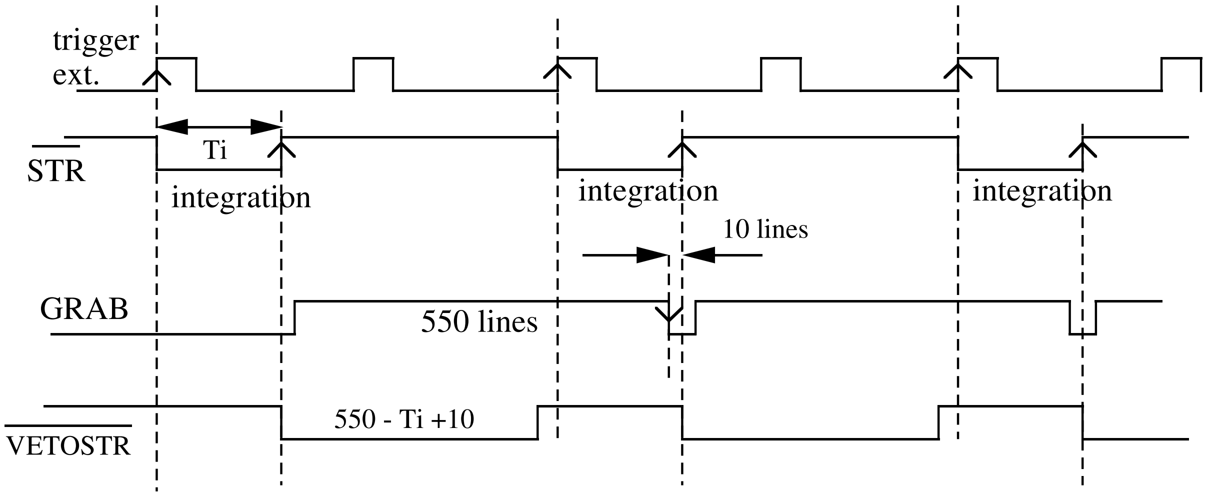

AQUISITION LOGIC MODULE
OEDCA
CCAEA
MUX
OEMA
WEMA
CRSA
OEDBA
VME Driver
Driver Cam.
OE
DBAP[7:0]
DBAP[15:8]
Mem 0
Mem 1
Mem 2
Mem 3
OE
CS0
CS1
CS2
CS3
WE
RGp
RGi
FR1
CRSA
DB[31:0]
DBAI[7:0]
DBAI[15:8]
0
15
16
31
ADDMA[16:0]
muxama
clock
write counter address
Mux
read counter address
D0
D17
I0
I16
D0-D16
FR1
load/count
vme BUS

Figure 2:



trigger
ext.
Ti
STR
integration
integration
integration
10 lines
GRAB
550 lines
550 - Ti +10
VETOSTR

Figure 3:

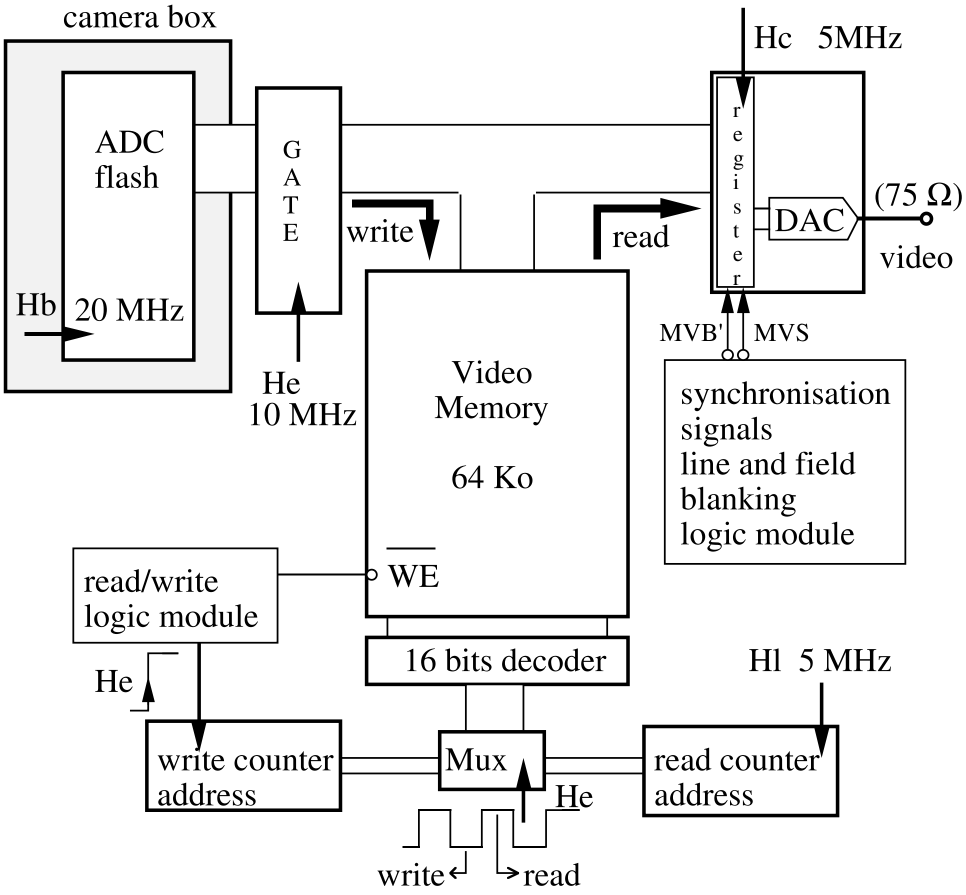

camera box
ADC
flash
Hb
20 MHz
G
A
T
E
He
10 MHz
write
read
Hc 5MHz
register
DAC
(75 Ω)
video
MVB'
MVS
synchronisation
signals
line and field
blanking
logic module
Video
Memory
64 Ko
WE
read/write
logic module
He
16 bits decoder
Hl 5 MHz
write counter
address
Mux
He
read counter
address
write
read

Figure 4:

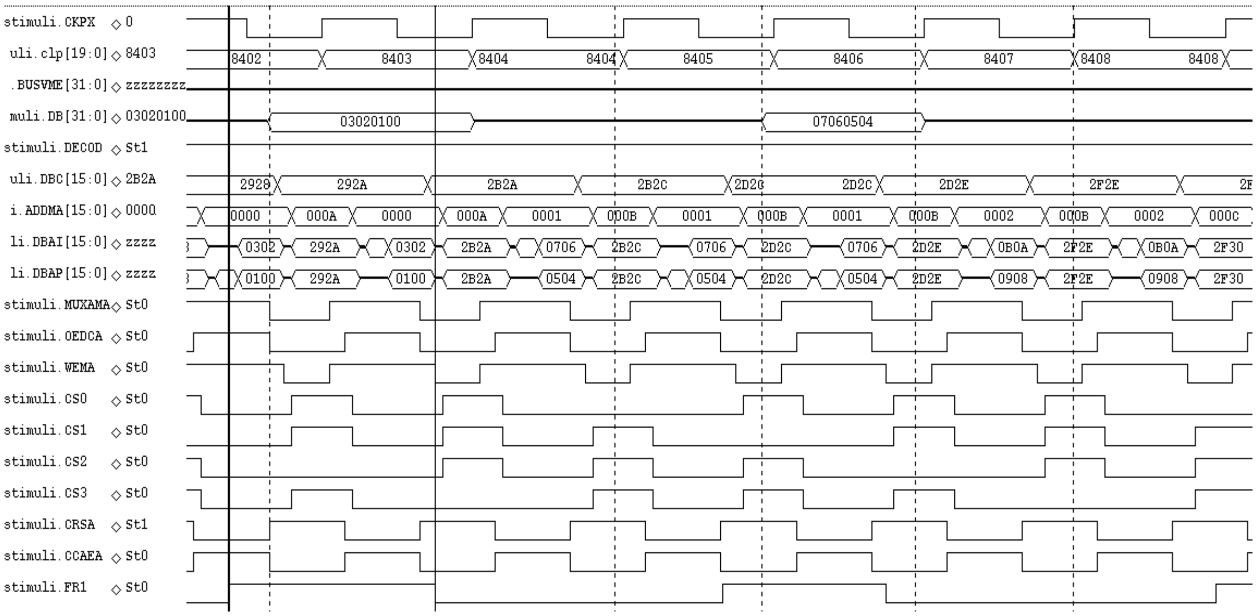
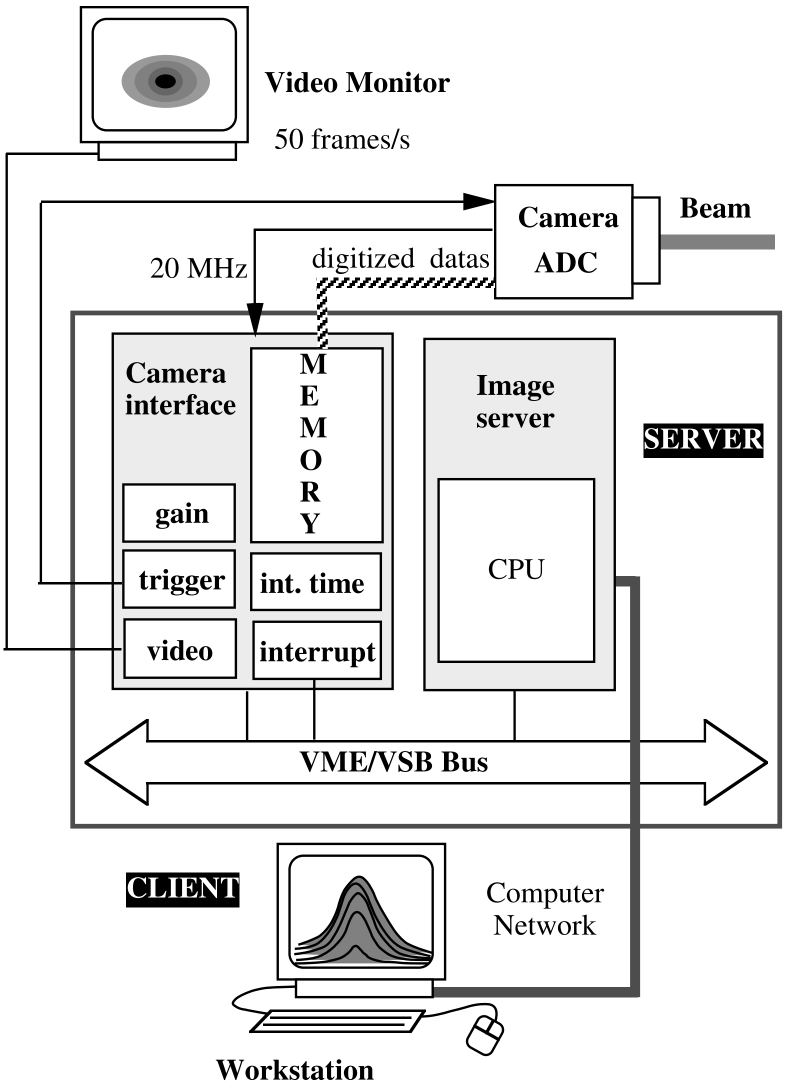

stimuli.CKPX 0
uli.clp[19:0] 8403
.BUSVME[31:0] zzzzzzzz
muli.DB[31:0] 03020100
stimuli.DECOD St1
uli.DBC[15:0] 2B2A
i.ADDMA[15:0] 0000
li.DBAI[15:0] zzzz
li.DBAP[15:0] zzzz
stimuli.MUXAMA St0
stimuli.OEDCA St0
stimuli.WEMA St0
stimuli.CS0 St0
stimuli.CS1 St0
stimuli.CS2 St0
stimuli.CS3 St0
stimuli.CRSA St1
stimuli.CCAEA St0
stimuli.FR1 St0

Figure 5:



Video Monitor
50 frames/s
Camera
ADC
Beam
20 MHz
digitized datas
Camera interface
M E M O R Y
gain
trigger
int. time
video
interrupt
Image server
CPU
SERVER
VME/VSB Bus
CLIENT
Computer Network
Workstation

Figure 6:

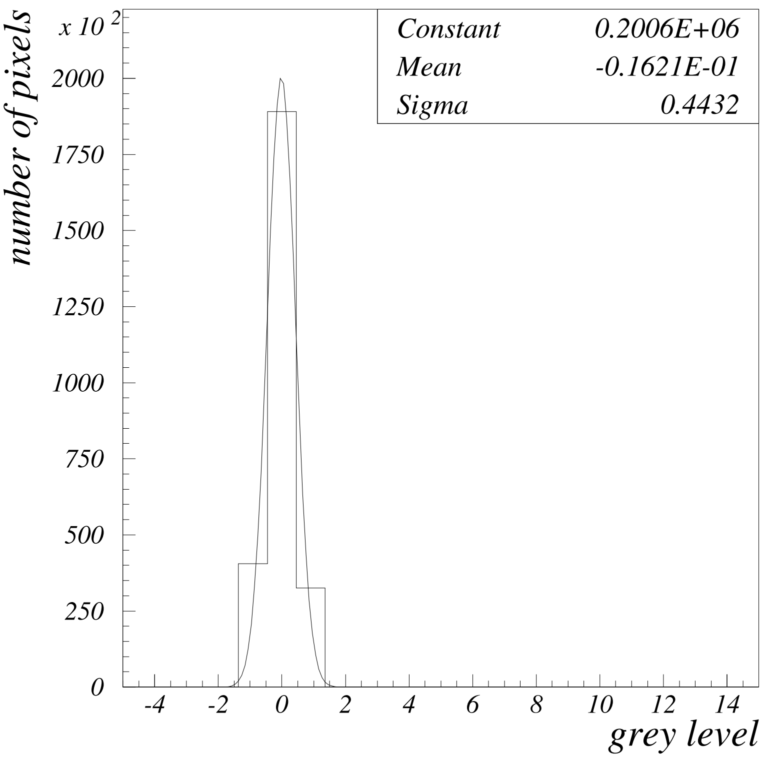

x 10 2
number of pixels
2000
1750
1500
1250
1000
750
500
250
0
-4
-2
0
2
4
6
8
10
12
14
grey level
Constant 0.2006E+06
Mean -0.1621E-01
Sigma 0.4432

Figure 7:

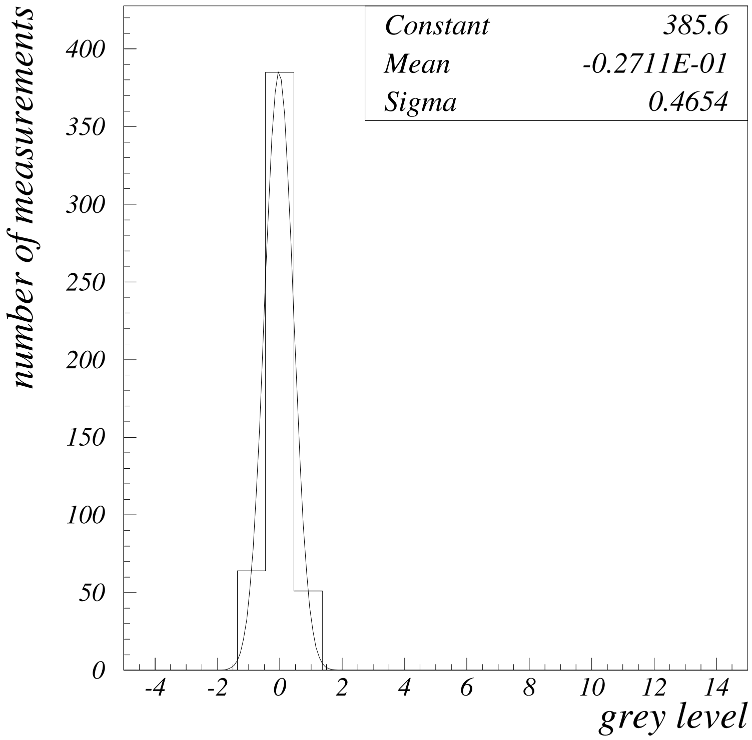
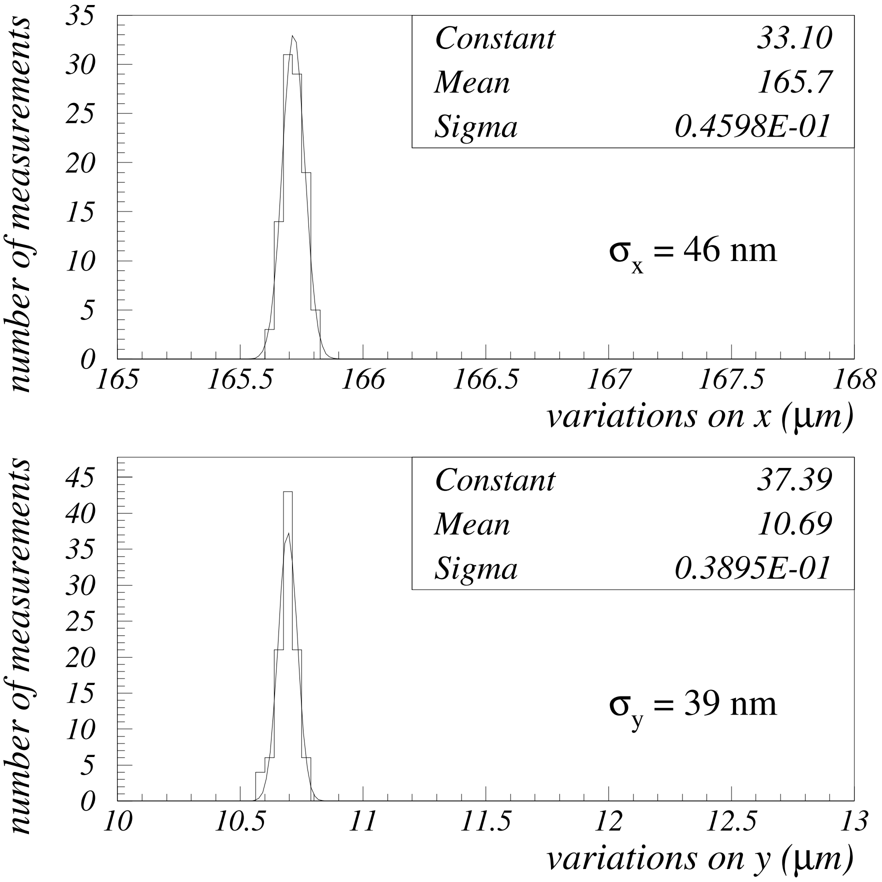
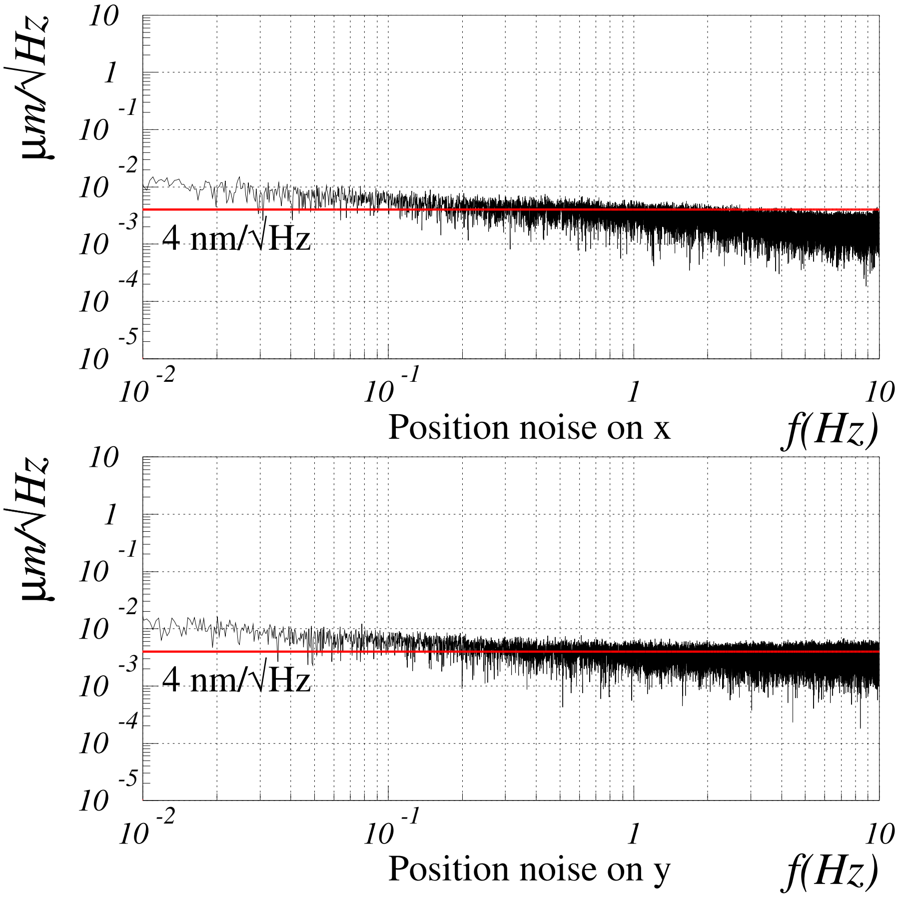

Constant 385.6
Mean -0.2711E-01
Sigma 0.4654
number of measurements
0
50
100
150
200
250
300
350
400
-4
-2
0
2
4
6
8
10
12
14
grey level

Figure 8:



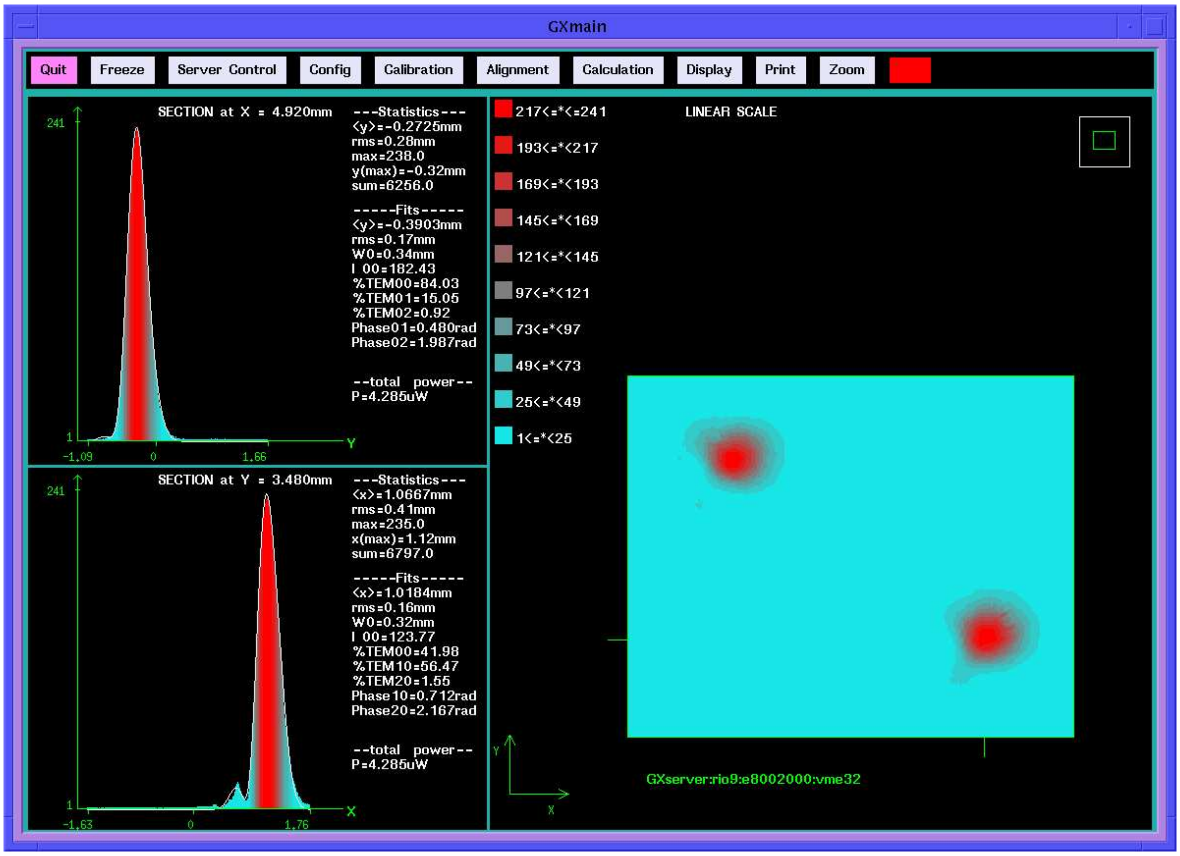
Constant 33.10
Mean 165.7
Sigma 0.4598E-01
σ_x = 46 nm
number of measurements
variations on x (μm)
Constant 37.39
Mean 10.69
Sigma 0.3895E-01
σ_y = 39 nm
number of measurements
variations on y (μm)

Figure 9:



$\mu m/\sqrt{Hz}$
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
4 nm/√Hz
10^{-2}
10^{-1}
1
10
Position noise on x
f(Hz)
$\mu m/\sqrt{Hz}$
10
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
4 nm/√Hz
10^{-2}
10^{-1}
1
10
Position noise on y
f(Hz)

Figure 10:



GXmain
Quit
Freeze
Server Control
Config
Calibration
Alignment
Calculation
Display
Print
Zoom
SECTION at X = 4.920mm
---Statistics---
<y>=-0.2725mm
rms=0.28mm
max=238.0
y(max)=-0.32mm
sum=6256.0
-----Fits-----
<y>=-0.3903mm
rms=0.17mm
W0=0.34mm
I 00=182.43
%TEM00=84.03
%TEM01=15.05
%TEM02=0.92
Phase01=0.480rad
Phase02=1.987rad
--total power--
P=4.285uW
241
1
-1.09
0
1.66
Y
SECTION at Y = 3.480mm
---Statistics---
<x>=1.0667mm
rms=0.41mm
max=235.0
x(max)=1.12mm
sum=6797.0
-----Fits-----
<x>=1.0184mm
rms=0.16mm
W0=0.32mm
I 00=123.77
%TEM00=41.98
%TEM10=56.47
%TEM20=1.55
Phase10=0.712rad
Phase20=2.167rad
--total power--
P=4.285uW
241
1
-1.63
0
1.76
X
217<=*<=241
193<=*<217
169<=*<193
145<=*<169
121<=*<145
97<=*<121
73<=*<97
49<=*<73
25<=*<49
1<=*<25
LINEAR SCALE
Y
X
GXserver:rio9:e8002000:vme32

List of Figures.