Double linear array CCD front–end

M. Prevedelli
mrp@fci.unibo.it

May 15, 2007

1 Introduction

This document describes a simple front–end interface circuit for Hamamatsu CCD linear arrays. The circuit is meant to be used for data acquisition after a monochromator in a pulsed system with a repetition rate of about 1 kHz. The signals are actually converted by a PCI ADC board and the front end just generates all the synchronization and sampling signals. The analog outputs can also be fed to an oscilloscope for monitoring and alignment.

2 CCD characteristics

A short description of the Hamamatsu S8377/S8378 linear arrays is useful for understanding both the S/N characteristics and the front–end requirements. The full data sheet is available both at the Hamamatsu web site and, hopefully, with this documentation. The arrays are available in 128, 256, 512 pixels of $500\times50\mu$m for the S8377 series and 256, 512 and 1024 pixels of $500\times25\mu$m for the S8377. A picture showing the spectral response from 200 to 800 nm can be found on the CCD data sheet.

The main feature of these arrays is that they include on chip CMOS digital driving logic and charge integrating amplifier, so, except for the analog output, all the inputs are TTL digital signals, and a single +5V supply voltage is required.

The digital signals are `VG, CLK, ST, EOS`. `VG` is simply a control voltage that sets the high (1 pF) or low (5 pF) gain for the charge amplifier. `CLK` is a clock signal that must be continuously running for proper operation. The maximum click frequency is 500 kHz. `ST` controls the integration: `ST` is sampled at the $1 \rightarrow 0$ clock transition; if at that time `ST` is 0 a readout–reset cycle is started and the analog output of the n pixels will be available at the analog output at the next $n0 \rightarrow 1$ clock transitions. Reading is a destructive operation and is actually the only way to reset the pixels. After reading each pixel charge accumulation restarts immediately. `EOS` will signal the end–of–scan by going to 0 at the $n+1$ clock $0 \rightarrow 1$ transition. `EOS` is actually ignored by the front–end.

The analog output voltage of a dark pixel will be about 1V while a saturated pixel will give a typical output voltage of 4.2V or 3.5V if in high or low gain mode respectively.

Reading the whole array at the maximum clock frequency requires $n \times 2\mu$s. If the arrays have to be read at 1kHz rate only the 128 and 256 arrays are usable. This circuit has been designed for interfacing 2 arrays S8377 of 256 pixels each but it will be easily adapted to other models.

3 S/N evaluation

The Signal–to–noise ratio and the dynamic range of the analog output should be estimated in order to choose a proper ADC. The read–out noise of the arrays are 150 μV and 500 μV rms for low and high gain respectively. Since the maximum signals are 2.5V and 3.2V (once subtracted the 1V offset), the maximum dynamic range are $\sim$ 16700 (low gain) and 6400 (high gain).

At low light levels the noise will be the read–out noise plus the shot noise on the dark current $I_d = 0.08$ pA. If n laser pulses are integrated on the CCD the charge increases by $8 \cdot 10^{-17}$ C/pulse i.e. 500 e/pulse. Fluctuations are then of the order of $0.7\mu\text{V} \cdot \sqrt{n}$ (low gain) or $3.5\mu\text{V} \cdot \sqrt{n}$ (high gain). It is then clear that the effect of the dark current can be neglected for reasonable integration times. Saturation due to dark current integration occurs after some 2.5 minutes of integration and can also be neglected.

At high light levels, the shot–to–shot noise is due, in the most favorable case, to $\sqrt{N_e}$ fluctuation in the number of photoelectrons N_e. Since saturation occurs at $N_e \simeq 8 \cdot 10^8$, the maximum S/N at high light levels is limited to about 1 part in 10^4.

4 Circuit description

The schematics of the CCD front–end are shown in Fig.(1). The circuit is fairly simple since all the digital logic is included in a single programmable chip (IC5). The most expensive and fragile components on the board are the CCD arrays, so all the CCD input and output signals have been buffered. IC2 is a dual op–amp that buffers the analog outputs. It must be a fast component since should settle to 0.005% of the final value in 2 μs to guarantee 14 bits readout at 500 kHz. All the digital inputs are routed through IC5. The +5V are generated by IC4. D1 protects the circuit against polarity inversions. VR1 is a 2.5V voltage reference that can be used to generate a 1V stable voltage with R2. By using insulated BNCs for the analog outputs, the 1V signal can be connected to the return so that a differential ADC board can remove the 1V offset and use the full dynamic range of the converter. In this case, however, connecting the output directly to an oscilloscope will short VR1. OG1 is the main system clock. It is divided by 8 for generating the CCD `CLK` signal. Its frequency is 3.54 MHz. IC5 takes care of everything else:

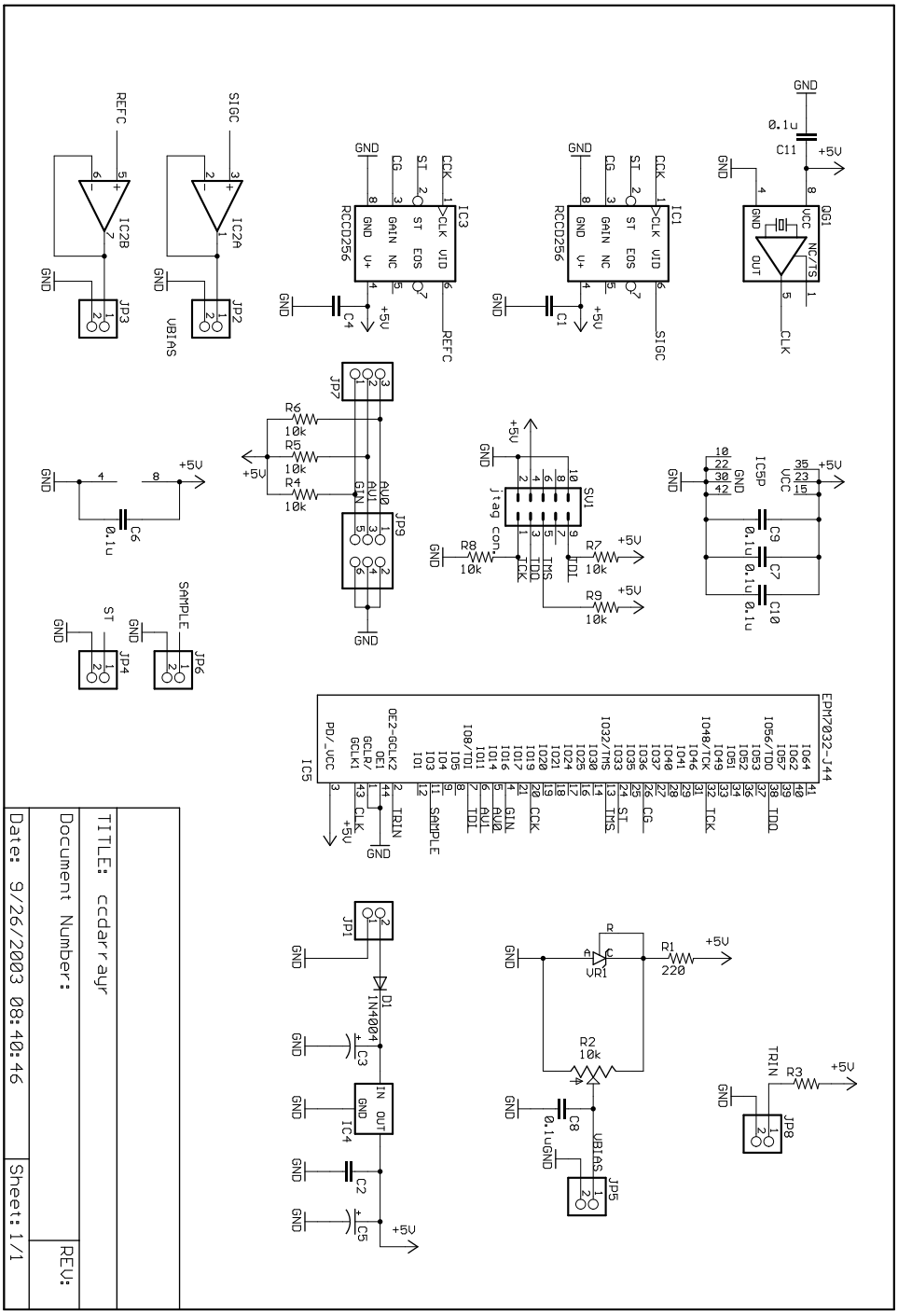

Figure 1: Schematics of CCD Front–end board.

1) synchronize the laser trigger at 1 kHz with the internal clock (i.e. generate the CCD `ST` signal;

2) generate the CCD `CLK` signal;

3) buffer the CCD `VG` signal;

4) generate a `SAMPLE` signal for the ADC board. The delay between the CCD `CLK` and `SAMPLE` can be adjusted in 1/8 `CLK` period steps.

5) allow for four different values of hardware average: by using the 2 TTL lines AV0 and AV1, only one trigger signal out of n will generate a readout cycle. Defaults for n are 1,20,50 and 100 but they can be easily changed by reprogramming IC5;

6) an external window discriminator can be added to the circuit. The TTL output of the external discriminator can be applied at the syncin pin of IC5. Internal logic generates an output TTL signal at syncout. Syncout is just a copy of syncin if not in average mode. Otherwise marks a whole hardware average as invalid if at least on pulse in the average in invalid.

The verilog (a hardware description language) code for IC5 is shown below. It is obvious which lines should be changed to modify the number n of hardware averages even if you do not know (and do not want to know) verilog.

The chip manufacturer gives for no cost a software (Maxplus, at `http://www.altera.com`) that can convert this code into a program that can be downloaded into IC5 by using a special interface (available at lens, but see Appendix I) connected to the PC parallel port.

```
// hamamatsu linear array driver
// mp 2007; added synchronizer for discrimination ff

module ccd(ckin,trin,gin,ave,ckccd,trccd,gout,sample,syncin,syncout);

input ckin,trin,gin;
input [1:0] ave;
input syncin;

output ckccd,trccd,gout;
output sample;
output syncout;

//gain in out: just a buffer
assign gout=gin;

// Pin assignmets
// ckin     = 43
// trin     =  2
```

```
// gin      =  4
// ave[0]   =  5
// ave[1]   =  6
// ckccd    = 20
// trccd    = 24
// gout     = 26
// sample   = 11
// syncin   = 41
// syncout  = 40

//input clock between 3.5 and 4 MHz
//generate ccd clock and card clock at input/8
//with both phases.
//Ccd clock and sample have separate pins so that rel. phases
//can be adjusted

wire[7:0] dq;
wire[2:0] nq;
wire r, inttrig;
assign inttrig=~trin;

wire ckaux;

lpm_counter cnt1 (.clock(ckin),.q(nq),.eq(dq));
defparam cnt1.lpm_width = 3;

assign ckccd=nq[2];

reg tr0,tr1;

//trigger:
wire p1,p2;
assign p1=dq[2];
assign p2=dq[6];

always@(negedge r or negedge tr1)
if(!tr1)
tr0=1'b0;
else
tr0=1'b1;

always@(posedge p2 or posedge p1)
if(p1)
tr1=1'b1;
else
if(tr0) tr1=1'b0;
```

```
assign trccd=tr1;

//average counter 3=no average    2=2 averages
//               1=5 averages
// 0=10 averages
// 1=5 averages
// 2=2 averages
// 3=0 averages
// but numbers can be changed below

‘define ave0  99
‘define ave1  49
‘define ave2  19
‘define avewidth 7

wire [‘avewidth-1:0] aq;
wire [3:0] dt;
wire v2,v1,v0;

lpm_counter cnt2 (.clock(~inttrig),.q(aq),.aclr(!tr1));
defparam cnt2.lpm_width = ‘avewidth;

lpm_compare cmp1 (.dataa(aq),.datab(‘ave0),.aeb(v0));
defparam cmp1.lpm_width = ‘avewidth;
defparam cmp1.one_input_is_constant = "YES";

lpm_compare cmp2 (.dataa(aq),.datab(‘ave1),.aeb(v1));
defparam cmp2.lpm_width = ‘avewidth;
defparam cmp2.one_input_is_constant = "YES";

lpm_compare cmp3 (.dataa(aq),.datab(‘ave2),.aeb(v2));
defparam cmp3.lpm_width = ‘avewidth;
defparam cmp3.one_input_is_constant = "YES";

assign dt[3]=inttrig;
assign dt[2]=v2;
assign dt[1]=v1;
assign dt[0]=v0;

mux mx(.sel(ave),.data(dt),.result(r));
defparam mx.width = 4;
defparam mx.widths = 2;

//sample
wire[9:0] q3;
```

```
wire ct3en;
reg ct3en;

always@(posedge ckin)
if(!trccd)
ct3en=1'b1;
else
if(q3[8])
ct3en=1'b0;

wire p3;
assign p3=dq[3];

lpm_counter cnt3 (.clock(p3),.q(q3),.aclr(!tr1),.clk_en(ct3en));
defparam cnt3.lpm_width = 9;

assign sample=p3 & ct3en;

// external discriminator sync
// syncin is sampled after 32 ccdck pulses after trin
// and stored in the jk flip-flop ffsync
// when in average mode a single 0 sets ffsync to 0 until readout
// ffsync is reset at pixel 240 during readout

wire[5:0] qau;
lpm_counter cntaux (.clock(ckccd),.q(qau),.aclr(trin),.clk_en(~qau[5]));
defparam cntaux.lpm_width = 6;

wire so;
assign syncout=so;
jkff ffsync (.j(syncin),.k(0),.clk(qau[5]),
             .clrn(~(q3[7] & q3[6] & q3[5])),
             .prn(1),.q(so));


endmodule
```

The pin connection table for IC5 can be derived from the schematics, but is reported here for convenience.

5 Construction

Construction presents no particular difficulties. Fig.(2) together with the following part list should be self explanatory. Just mind that the two CCD arrays (IC1 and IC3) must be mounted on the solder side.

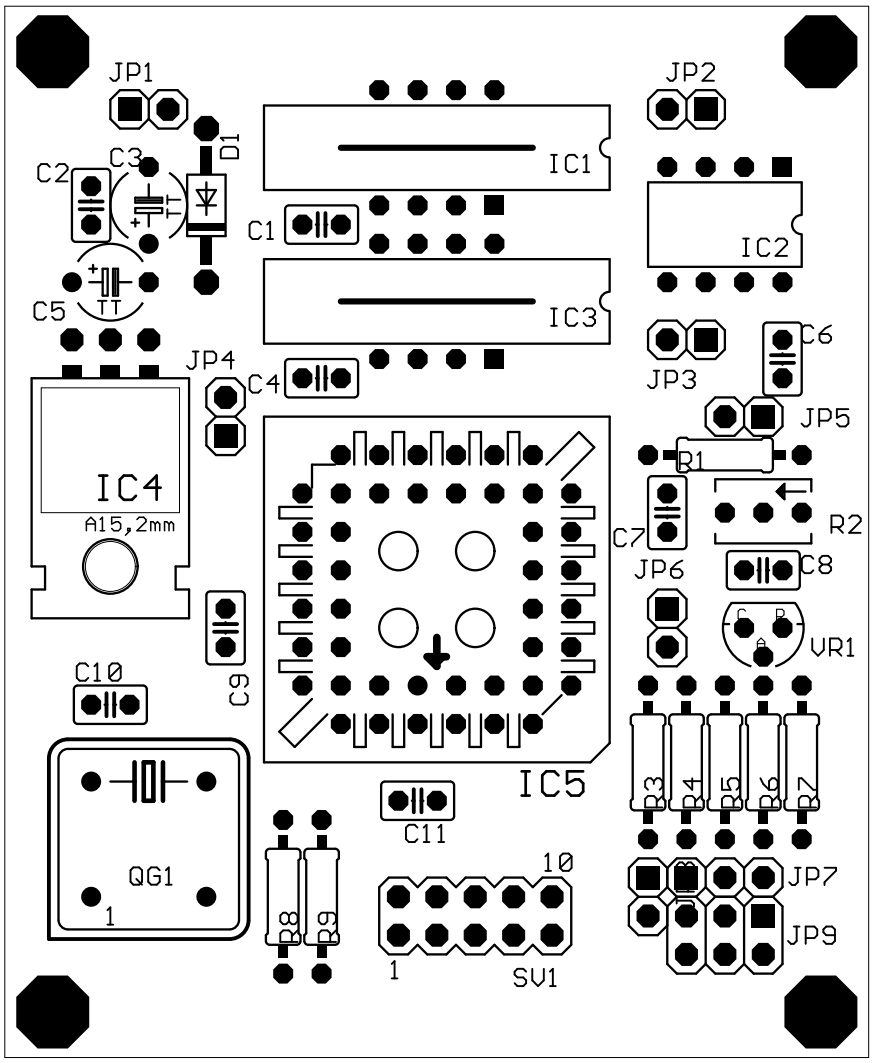

Figure 2: Top view of the CCD Front–end board (2:1 scale).

signal	pin	description
trin	2	Trigger input
gin	4	Gain input (from PC)
ave0	5	Hardware average selection (from PC)
ave1	6	Same as above
sample	11	ADC board trigger output
ckccd	20	CCD `CLK` signal
trccd	24	CCD `ST` signal
gout	26	CCD `VG` signal
ckin	43	main clock input (3.54 MHz)
syncin	41	discr. in
syncout	40	discr. out

Table 1: Connection table for IC5.

Components List

Qty	Value	Parts
1	OP270	IC2
1	7805	IC4
1	TL431CLP	VR1
1	3.54 MHz Quartz	QG1
9	0.1u	C1, C2, C4, C6, C7, C8, C9, C10, C11
1	1N4007	D1
2	4.7u	C3, C5
7	10k	R3, R4, R5, R6, R7, R8, R9
1	220	R1
1	10k	R2
1	EPM7032-J44	IC5
2	S8377-256	IC1, IC3

6 I/O Interface

We have chosen to use a Adlink `http://www.adlinktech.com` Daq2010 as ADC board. The daq2010 is an excellent piece of hardware with horrible software and documentation. It is used here because has also 24 digital I/O lines (only 3 are used to set gain and average) and because is a reasonably fast, simultaneous sampling, board: at the external trigger up to 4 channels (only 2 used) are simultaneously sampled and later converted at a maximum rate of 2 MHz with 14 bits accuracy. Analog inputs can be differential and full scale range can be bipolar or unipolar and range from 1.25V to 10V.

The board is connected to the front end through 2 insulated BNC (for the analog arrays outputs) and a 9 poles D connector, for the 4 digital signals (`sample, ave0, ave1, gin`) and ground.

The only other connection required is the TTL trigger input, through a BNC connector.

7 Pulse Discriminator

Appendix I

The hardware interface between the PC parallel port and the JTAG connector on the PCB board (SV1, in the schematics) is just two connectors, a 74245 buffer and few resistors. It fits on a $50x50$mm single face PCB. It is simple enough to be build also with flying wires. In fig.(3) the schematics are shown. Fig.(4) and fig.(5) show the silk–screen and solder side image of the board we have been using.

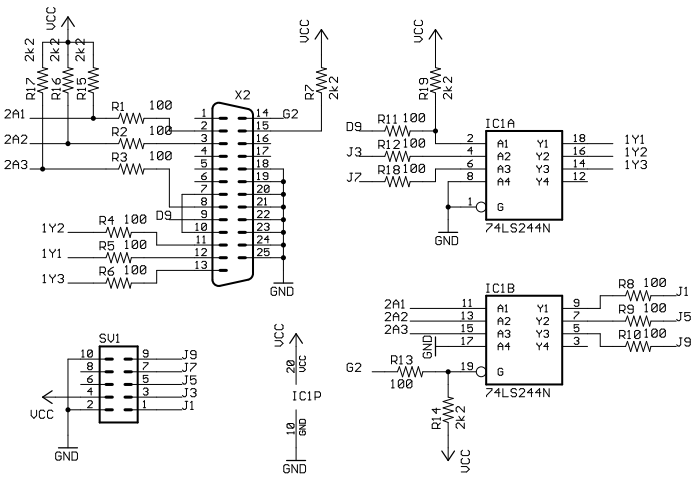

Figure 3: Schematics of the PC–JTAG interface.

Note that all the series resistors are 100Ω while all the pull–ups are 2.2kΩ. The 25 pins sub-D male connector can inserted directly in the PC parallel port connector while for the JTAG connector a 10 wire flat cable is required. Be careful to connect the #1 pin on the adapter with the #1 pin (use polarized connectors if possible). The interface is powered by circuit to be programmed via the flat cable.

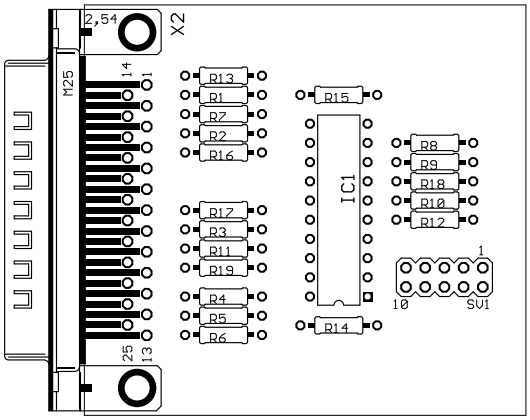

Figure 4: Top view of the PC–JTAG interface.

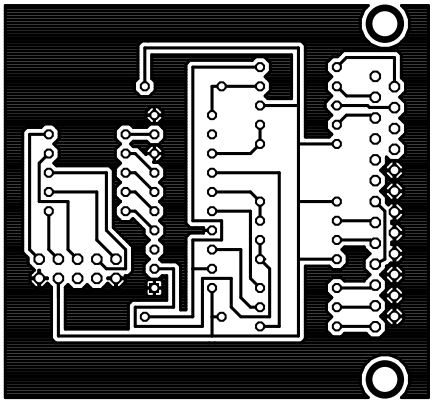

Figure 5: Solder side in scale 1:1 of the PC-JTAG interface.