Double linear array MCT front–end

M. Prevedelli
mrp@fci.unibo.it

December 9, 2003

1 Introduction

This document describes a front–end interface circuit for a double array of 32 MCT sensors. Since each sensor has an independent output, 64 preamplifiers, gated–integrator and sample–and–hold are required. Each channel is kept as simple as possible since we need a lot of them. Two 32–to–1 multiplexers are then used to generate 2 CCD–like serial outputs for more convenient monitoring and acquisition. Some digital logic is also required to synchronize sampling on an external trigger coming from the laser amplifier (the MCT will be used with a pulsed system running at 1 kHz).

2 MCT characteristics

Some of the electrical properties of the MCT sensors must be considered in order to design the preamplifier.

Each sensor has an area A of $0.2 \times 0.05\, cm^2$. The detectivity is guaranteed to be $D^* > 2 \times 10^{10}\mathrm{cm}\sqrt{\mathrm{Hz}}/\mathrm{W}$ and the responsivity $\eta \simeq 2000\mathrm{V/W}$. The maximum intensity for linear operation is around $I_m \simeq 1\mathrm{mW/mm}^2$. Typical sensor resistance R and time constant τ are $R = 80 \sim 150\Omega$ and $\tau = 1\mu\mathrm{s}$. The bias voltage V_b can range from 1.5 to 2V. The bias current varies from 10 to 25 mA. Note that the total bias current is of the order of 1 A and that the electrical power dissipated in the dewar due to the bias amounts to 2W.

From the area A and the detectivity $D*$ we obtain the N.E.P. as

$$\text{N.E.P.} = \frac{\sqrt{A}}{D^*} = 5 \times 10^{-12}\,\mathrm{W}/\sqrt{\mathrm{H}z} \tag{1}$$

and, through the responsivity η, the voltage noise spectral density at the sensor output.

$$V_n = \eta\,\text{N.E.P.} = 10\,\mathrm{nV}/\sqrt{\mathrm{Hz}} \tag{2}$$

It is useful to compare V_n, which hopefully depends on the fluctations of black body radiation, with thermal and shot noise. Thermal noise at 70K for a 100 Ω resistor is

$$V_t = \sqrt{4RkT} = 0.6\,\mathrm{nV}/\sqrt{\mathrm{Hz}} \tag{3}$$

while the shot noise for a 10 mA bias current I_b is

$$V_s = R\sqrt{2eI_b} = 5.7\,\mathrm{nV}/\sqrt{\mathrm{Hz}} \tag{4}$$

It is then clear that while V_t is negligible, some attention should be paid to the bias source because some technical extra noise could rise V_s above V_n.

The maximum signal is

$$V_M = \eta A I_s = 2\,\mathrm{V} \tag{5}$$

V_M integrated over τ gives then a signal $S_M = 2 \times 10^{-6}$ Vs

The typical energy E per pulse on each sensor should be of the order of 100 fJ. The integrated pulse has then an area of

$$S = \eta E = 2 \times 10^{-10}\,\mathrm{V}s \tag{6}$$

The preamplifier must then have

1) a bandwidth of about 1 MHz in order not to limit the sensor response;

2) a gain of at least 100[1]

3) AC coupled input;

4) a noise below V_n;

The maximum S/N in a single pulse is of the order of

$$S/N = \frac{S_M}{V_n}\sqrt{\tau} = 200000 \tag{7}$$

The typical S/N in a single pulse however is 10000 times less i.e. only 20.

Since the data acquisition rate does not require a fast ADC (a sampling rate of 100kHz will be enough) it is possible to work with a cheap 16 bits ADC card.

3 Circuit description

3.1 Preamplifier and gated integrator

The schematics for a single channel is shown in Fig.(1). IC2 (an AD797A) is the preamplifier, with a gain of 101, a bandwidth around 1 MHz and a noise level of 1 $\mathrm{nV}/\sqrt{Hz}$. IC1A (a precision FET–input dual op–amp, AD712K) and the 2 analog switches IC3A and IC3D (ADG441, low leakage; a normal DG211 will

[1] This is the highest gain that can be achieved with 1 MHz bandwidth by a single low noise amplification stage.

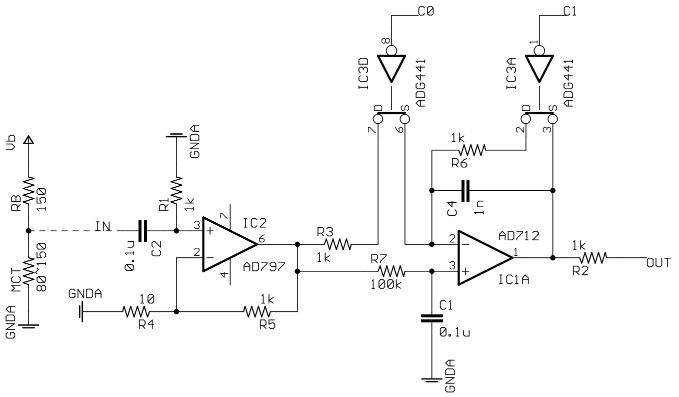

Figure 1: Schematics of MCT preamplifier and gated integrator.

probably spoil the circuit) form the gated integrator. Since there are 2 op–amps in IC1 and 4 switches in IC3, only another AD797 is needed to build another channel. A single euro–card board will fit 8 channels. C_0 is the hold/integrate switch. C_1 is the integrate/reset switch. For integration C_0 must be closed and C_1 open. Opening C_0 then puts the circuit in hold mode where the integration result is "stored" in C4. Closing C_1 finally resets the integrator. Note that in reset-mode the integrator is actually an inverting amplifier.

The main source of error for the integrator is the amplified input voltage of IC2. To this error contribute both the voltage offset V_{of2} and the bias current I_{b2} (flowing through R2) of IC2 Since the typical value for V_{of2} is 25 μV and I_{b2} is an unusually large 0.25 μA, I_{b2} dominates for R2 greater than 100 Ω. On the other hand R2 should be much greater than the sensor resistance in order not to loose signal. The actual impedance seen by the sensor will be set by the bias resistor RB.

With R2=1KΩ the total error is typically 25 mV at the input of the integrator. With an integration time of 5μs the offset error at the output of the integrator is then 130 mV while our typical signal is $V_o = 100 \times 10^6 \mathrm{s}^{-1} \times S = 20$ mV. The offset can be sampled and subtracted via software but for a hardware solution a possibility is to subtract the average value of the offset via the passive integrator formed by R5 and C3. Otherwise R2 can be reduced to 100ω, R5 omitted and C3 replaced by a jumper. The error in the signal amplitude introduced by R5 and C3 is reduction of the order of (R3·C4)/(R5·C3)=10^{-4}.

In hold mode errors are due to the leakage current I_l through C_0 and C_1 and to IC1 input bias current I_b1. Typical values for I_l and I_b1 are 10 and 25 pA respectively. Leakage current should then be around 50 pA. The typical charge on the integration capacitor C1 is $C1 \times 20\mathrm{mV} = 20pC$ per shot. Discharge time can be then of the order of 0.5 s and leakage is totally negligible in 1 ms. Be aware anyway that I_b1 roughly doubles every 10 C° and the typical values are given for a chip temperature of 25 C°. Hardware averages of many pulses are then possible since “signal” charge increases as 20 nA current (20pC × 1 kHz) while leakage charge rate is 50 pA, corresponding to a pasitic resistance of 400 GΩ. leakage from PCB tracks or C1 are probably larger. More serious is the error introduced by charge injection from C_0: due to capacitative coupling, every on/off cycle injects some charge into C1. IC3 is a low charge injection device and is specified for 1 pC typical (6 pC maximum). For comparison, a DG211 is specified for 20 pC typical. Charge injection then produces a typical 4% error and this is not reduced by averaging more pulses.

Thermal drifts of these error sources are much smaller so it is possible to acquire every few minutes a “dark” sequence (there is no need to block the laser, under software control the circuit can shift the integration gate away from the laser pulse, see below) and subtract it from the signal.

Typical integration, hold and reset times will be 5 μs, 800 μs and 200 μs respectively, in case of no averaging. For an n pulses average the hold time will increase to about n ms. The maximum value for n is limited to 1024.

The complete schematics together with the part list for an 8 channels board is shown in Appendix I. It would be nice to implement a real boxcar integrator, integrating just after or before the signal the inverted background for the same time. This would require at least another op–amp and an analog switch per channel. With 64 channels this will probably increase too much the complexity of the circuit.

3.2 Bias source

Each sensor requires a bias current ranging from 10 to 20 mA. A bias resistor Rb is required to increase the input impedance seen by the sensor. Rb should be comparable with or larger than the sensor resistance. On the other hand increases the required voltage and power dissipation. A reasonable compromise is to set Rb=200Ω and use a 5V supply capable of current of 1.5A. On each 8 channel board (see Fig.(6)there is an independent voltage regulator build using an LM317 (IC 17) and a TL431 (VR1). The bias current can then be “almost independently adjusted” since every group of 8 sensors has its own voltage regulation trimmer (R58). The main problem with the bias current is to keep the voltage noise below V_n. A typical voltage reference has a noise spectral density around $100\,\mathrm{nV}/\sqrt{\mathrm{Hz}}$ plus some $1/f$ noise at low frequency and technical noise at 50 Hz and harmonics. A factor 2 attenuation attenuation is provided by the voltage divider formed by Rb and the sensor. Additional filtering is provided by a low pass LC filter formed by L1 and C57 and by the high pass filter at the input of every channel. In any case the voltage noise of the bias surce should be checked with a 150Ω resistor as a fake sensor element and the 5V power supply should be a linear (i.e. not switching) model, and maybe not the same used to power digital logic on the control board.

3.3 Output multiplexers

This section uses four 16 to 1 analog multiplexers (ADG406, IC2–3 IC5–6 in Fig.(9)) to build two 32 to 1 multiplexers in order to have two separate outputs for the signal and reference channels. After the multiplexers there are two op–amps (AD797, IC1 and IC4 in Fig.(9)) in non–inverting configuration to be used as an extra gain stage and buffer. The AD797 is not used here for its low noise characteristics but for the gain–bandwidth product.

Two different gains (2 and 20) can be selected via software using half of an ADG441 (IC7) to change the feedback resistors on IC1 and IC4. The exact value of the high gain depends also on the "on" resistance of the switches. The high gain should boost the integrator output signal for a single pulse V_o up to 800 mV.

Note that the values before the multiplexers are available at two 32 pins connectors (JP5 and JP6 in the schematics in Fig.(9) in Appendix I)

3.4 PLL and digital section

All the synchronization signals are derived from the laser trigger at 1 kHz using a PLL (IC9 in Fig.(10), a "classic" 74HC4046) locked on the 1000*th* harmonic of the trigger signal. The PLL bandwidth has been set in the 100 Hz range. See the 74HC4046 data sheet and adjust R19,R20 and C20 to change it. The 1/1000 digital divider, the driving signals for the C_0 and C_1 switches and the `sample` signal for the ADC board are generated with a CPLD logic of the Altera MAX7000S family, an EPM7064, IC8 in Fig.(10). The Verilog code for the logic is given below. The chip can be programmed on board by using a hardware interface described in Appendix III via the parallel port of a PC using a public domain software from Altera. The programming connector is SV1 in Fig.(10). The main operating parameters (see below) can be changed by editing some constants at the beginning of the code and by reprogramming the chip.

```
// mct array driver

module mct(ckin,mode,ad,ad4n,c0,c1,sample,gc,ckout,trgout);

‘define startint   10’d2
‘define stopint    10’d7
‘define startdummy 10’d100
‘define stopdummy  10’d105
‘define muxdelay   10’d2
‘define muxtime    5’d20
‘define resettime  10’d800
‘define average    10’d100
‘define sensnum    5’d16

input ckin;
input [2:0] mode; // operating modes:
```

```
                  // 000 = normal
                  // 001 = average gain=2
                  // 010 = average gain=20
                  // 011 = background normal
                  // 100 = background average gain=2
                  // 101 = background average gain=20
                  // 110 = pseudo boxcar
                  // 111 = integrator off
output [4:0] ad;
output ad4n;
output c0,c1; //integrator control
output sample;
output gc;    //gain control
output trgout; //start of mux sequence
output ckout; //PLL output

//PLL divider
wire [9:0] mq;

lpm_counter pllcnt (.clock(ckin),.q(mq));
defparam pllcnt.lpm_width = 10,
         pllcnt.lpm_modulus = 1000;

//phase adjustable ckout
reg ckout;
always@(posedge ckin) ckout=(mq==0) ? 1 : 0;

//address Prescaler
wire adclk;
wire [4:0] prq;
reg adrst;

lpm_counter prcnt (.clock(ckin),.cout(adclk),.aclr(adrst),.q(prq));
defparam prcnt.lpm_width = 5,
         prcnt.lpm_modulus = ‘muxtime;

//address counter
wire [5:0] adq;
reg ld;
wire adcken;
wire [6:0] prset;

assign prset[5:0] = ‘sensnum;
assign prset[6] = 1;

lpm_counter adcnt (.clock(adclk),.clk_en(adcken),.data(prset),
                   .aload(ld),.aclr(adrst),.q(adq));
defparam adcnt.lpm_width = 6;

assign ad4n=~adq[4];
assign ad=adq[4:0];
assign adcken=~adq[5];

//gain
reg rgc;
always@(posedge ckin)
        case(mode)
          0,2,3,5,6,7 : rgc=0;
          1,4         : rgc=1;
        endcase

assign gc=rgc;

//average counter
wire avcout;
reg ensample;

lpm_counter avcnt (.clock(ckout),.cout(avcout));
defparam avcnt.lpm_width = 10,
         avcnt.lpm_modulus = `average;

always@(posedge ckin)
        case(mode)
          0,3,6   : ensample=1;
          1,2,4,5 : ensample=avcout;
          7       : ensample=0;
        endcase

//phase adjustable sample
reg sample;
always@(prq) sample=(prq==5) ? (adcken & ~adrst & ensample) : 0;


//C0, C1 assume that with 0 switch are closed
//C0=run/hold, C1=integrate/reset
reg c0,c1;
reg sig;

//trgout signal
assign trgout=adcken & ~adrst & ensample;

always@(negedge ckin) begin
```

```
if(mode==7) begin
   c0=0;
   c1=0;
   adrst=0;
   ld=1;
end
else begin
   ld=0;
   if(sig)
      case(mq)
        `startint : begin
                      c0 = 0;   // run
                      c1 = 1;   // integrate
                    end
        `stopint  : c0 = 1;   // hold
        `stopint+`muxdelay : if(ensample) adrst=0;  // start mux
        `resettime : if(ensample) begin
                        adrst=1;   // reset mux
                        c1=0;      // reset
                        case(mode)
                          0,1,2 : sig=1;
                          3,4,5 : sig=0;
                          6     : sig=~sig;
                        endcase
                     end
      endcase
else
   case(mq)
     `startdummy : begin
                     c1 = 1;   // integrate
                     c0 = 0;   // run
                   end
     `stopdummy  : c0 = 1;   // hold
     `stopdummy+`muxdelay : if(ensample) adrst=0;  // start mux
     `resettime : if(ensample) begin
                     adrst=1;   // reset mux
                     c1=0;      // reset
                     case(mode)
                       0,1,2 : sig=1;
                       3,4,5 : sig=0;
                       6     : sig=~sig;
                     endcase
                  end
   endcase
end
end
```

```
endmodule
```

The connection table for the CPLD is shown below in table(1).

signal	pin
ad0	16
ad1	17
ad2	18
ad3	19
ad4	20
ad4n	21
c0	25
c1	24
ckout	41
ckin	43
mode0	9
mode1	12
mode2	16
gc	28
sample	11
trgout	8

Table 1: Connection table for the CPLD.

3.5 Modes of operation

The circuit has eight different modes of operation. They can be selected either with switches on the front panel (connected to JP14, JP17 and JP18 in Fig.(10)) or via software by programming three TTL input lines (JP15, JP19 and JP20 in Fig.(10) reaching the CPLD. The eight modes are:

0 Normal mode: the gated integrators are operated at 1 kHz. After every trigger pulse there is an integration phase, hold, multiplexing and reset. The sampling signals for the ADC are generated for every trigger pulse. The pulse width and position with respect to the trigger can be set with 1μs resolution. The multiplexer gain is set to 20. This is supposed to be the default acquisition mode.

1 Average mode: the integrators will be reset and ADC the sampling signals generated only every n pulses, while integration and hold phases will be repeated for each pulse i.e. n shots will be averaged before readout. The value of n is hard coded in the CPLD and, by reprogramming the chip can take any value from 2 to 256. The default is 10. The multiplexer gain is set to 2.

2 Same as mode 1 but with multiplexer gain set to 20.

3 Background mode: same as mode 0 but, in order to measure the offset of the integrators, an acquisition cycle will be performed gating the integrators away from the laser pulses. The idea is that the software could perform a background acquisition cycle every few minutes or so, store the offsets ad subtract them from signals.

4 Same as mode 1 but for in background acquisition (see mode 3)

5 Same as mode 2 but for in background acquisition (see mode 3)

6 Pseudo boxcar: alternates a mode 0 and a mode 3 acquisition.

7 Integrator off: integrators are permanently held with $C0$ and $C1$ closed. The multiplexers are also blocked to transmit always the signal of the ith sensor. The value of i is hard coded in the CPLD and can be changed by reprogramming it. The default is $i = 16$ (pixels in the middle of the array). It can be used for diagnostics (i.e. have a look at the output and trigger signal on a dual channel scope to see where the integrator should be gated) or alignment. Multiplexer gain is 20.

Some constants at the beginning of the file control the main operation parameters. Just a simple explanation of the syntax: the line

```
'define startint   10'd2
```

defines `startint` as a 10 bits constant expressed in decimal (`10'd`) whose value is 2, which is the only number that should be changed, if needed. The meaning of the constants is quite obviuos, anyway, not to leave anything to immagination:

- `startint` integration start time (in μs, after the trigger pulse rising edge plus 0.5 μs, min=0, max=999).

- `stopint` integration stop time. Units and limits as above. Mind that a value of `startint` of 998 and of `stopint` of 8 means a 10 μs integration starting *before* the trigger pulse and is perfectly ok.

- `startdummy` integration start time for background acquisition.

- `stopdummy` integration stop time for background.

- `muxdelay` delay between end of integration and beginning of multiplexing (in μs).

- `muxtime` multiplexing time (in μs, max=32, default=20). It is also easy to change the delay between multiplexer switching and `sample` rise time. Have a look at the code. The default is 5 μs.

- `average` number of averages in modes 1 and 2 (max=1024)

- `sensnum` sensor observed in mode 3 (max=32)

I/O Connectors

The input signals are:

- sensors inputs: 2× 41 pins Canon Connectors
- trigger input: BNC (TTL logic levels)
- mode selection: 3 TTL logic inputs. It would be nice to use a sub–D 9 poles connector for digital I/O with the PC.

while for the outputs we have:

- multiplexed outputs: 2×insulated BNC (the ADC board should operate in differential mode.
- `sample` output: a TTL logic pulse to trigger a 2 channels scan on the ADC board. Uses the sub–D connector.
- `trgout` output: a TTL logic signal for triggering a scope or an ADC board. A rising edge will signal the start of a multiplexing cycle. Can use the sub–D connector if required by the ADC board or its own BNC if used for the scope. It does not work in mode 7 since there are no mux cycles. Use the 1 kHz trigger signal from the laser in this case.

Other parts that could optionally find a place on the front panel are

- the LED indicating the PLL lock condition;
- switches for mode manual mode selection;
- CPLD programming connector SV1 for avoiding opening the box if reprogramming is needed.
- trimmers for bias setting;

Appendix I: Complete schematics and part list

The full schematics for a single 8–channels board are shown in the following in Figs.(2–6). The board silkscreen is shown in Fig.(7). A total of 8 boards are needed for 64 channels. The part list for a single board is given in the table below.

Part list for the 8 channels board

Qty	Value	Parts	Notes
4	AD712K	IC3, IC7, IC11, IC15	Low bias current op–amp
8	AD797A	IC2, IC4, IC6, IC8, IC10, IC12,	

Qty	Value	Parts	Notes
		IC14, IC16	Low noise op–amp
4	ADG441	IC1, IC5, IC9, IC13	Low leakage quad. switch
1	LM317	IC17	
1	TL413	VR1	2.5V Voltage reference
1	H15ML	X1	15 Poles DIN41612 male conn.
1	2x8PINHD	JP1	Input Connector
1	2x4PINHD	JP2	Output Connector
41	1kΩ	R1–4, R7–11, R14–18, R21–25, R28–32, R35–39, R42–46, R49–53, R56–57	
8	100kΩ	R5, R12, R19, R26, R33, R40, R47, R54	
8	10Ω	R6, R13, R20, R27, R34, R41, R48, R55	
8	10kΩ	R58	Trimmer
1	200Ω	RN1	8x200Ω Resistor network
1	100μH	L1	
1	vk200	L2	
50	0.1μF	C1–2, C4–8, C10–15, C17–21, C23–29, C31–36, C38–43, C45–50, C52–56, C58, C60	
8	1nF	C3, C9, C16, C22, C30, C37, C44, C51	
2	47μF	C57,C59	

Schematics and silkscreen for the control board are shown in Figs.(8–10 and Fig.(11)

Part list for the 8 channels board

Qty	Value	Parts	Notes
2	AD797	IC1, IC4	
4	ADG406	IC2–3, IC5–6	16 to 1 analog switch
1	ADG441	IC7	
1	EPM7064	IC8	Altera MAX7000S CPLD
1	74HC4046	IC9	VCO+PLL
1	LED	D1	
1	H15ML	X1	15 Poles DIN41612 male conn.
11	1x2PINHD	JP1–2, JP13–21	
8	2x4PINHD	JP3–4, JP7–12	
2	2x16PINHD	JP5–6	
1	2x5PINHD	SV1	JTAG prog. connector

Qty	Value	Parts	Notes
12	10kΩ	R1, R3, R5–6, R8, R10–17	
2	1kΩ	R2, R7	
2	470Ω	R4, R9	
1	2kΩ	R18	
1	1.5MΩ	R19	
1	67kΩ	R20	
2	200kΩ	R21, R22	
18	0.1μF	C1–18	
1	100pF	C19	
1	1μF	C20	

Appendix II: Lock–i vs. Boxcar

Sooner or later I'll write this. Not now however.

Appendix III: JTAG Interface

The hardware interface between the PC parallel port and the JTAG connector on the PCB board (SV1, somewhere in the schematics) is just two connectors, a 74245 buffer and few resistors. It fits on a 50x50mm single face PCB. It is simple enough to be build also with flying wires. In fig.(12) the schematics are shown. Fig.(13) and fig.(14) show the silkscreen and solder side image of the board we have been using.

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.

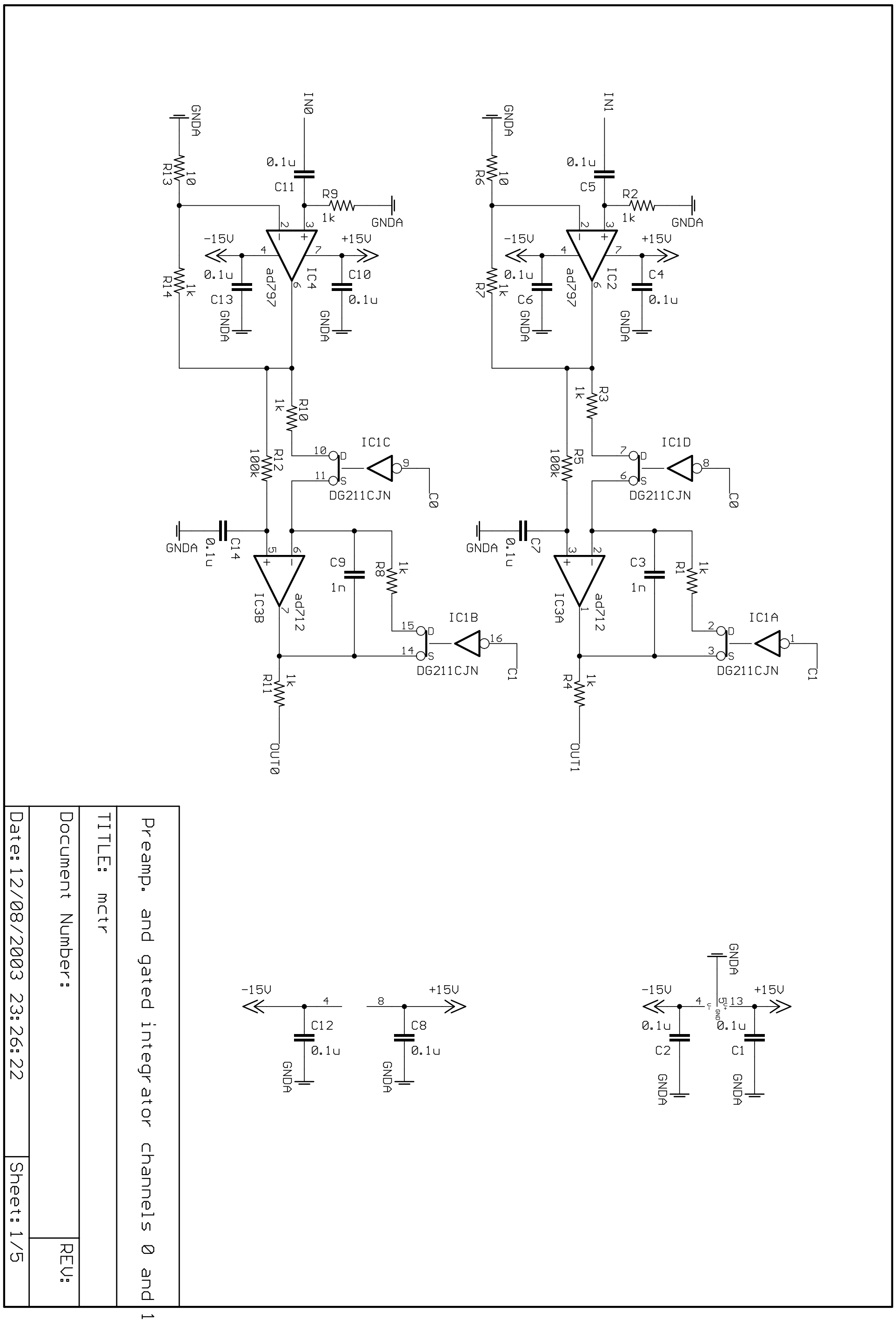

Figure 2: Detector channels 0 and 1.

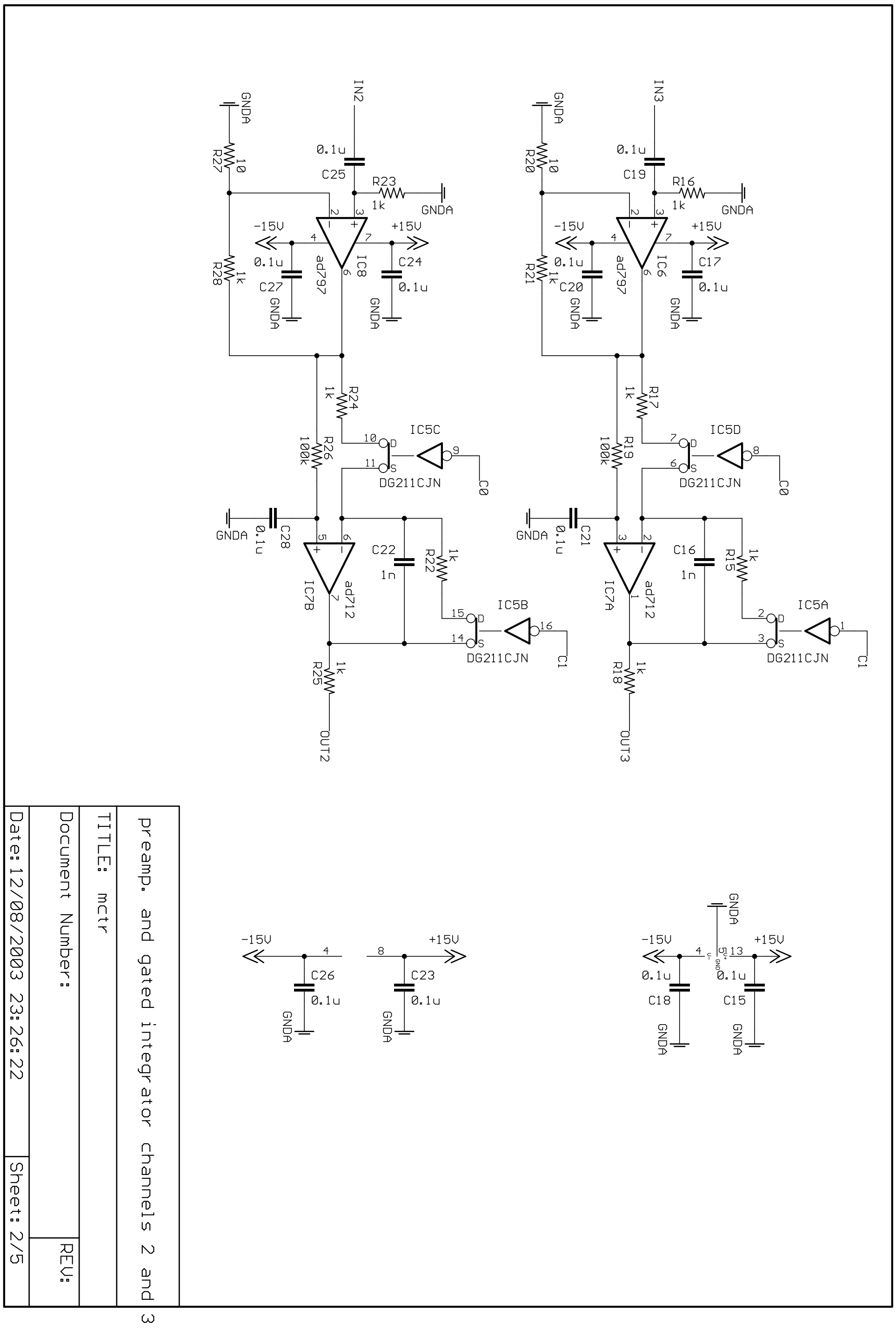

Figure 3: Detector channels 2 and 3.

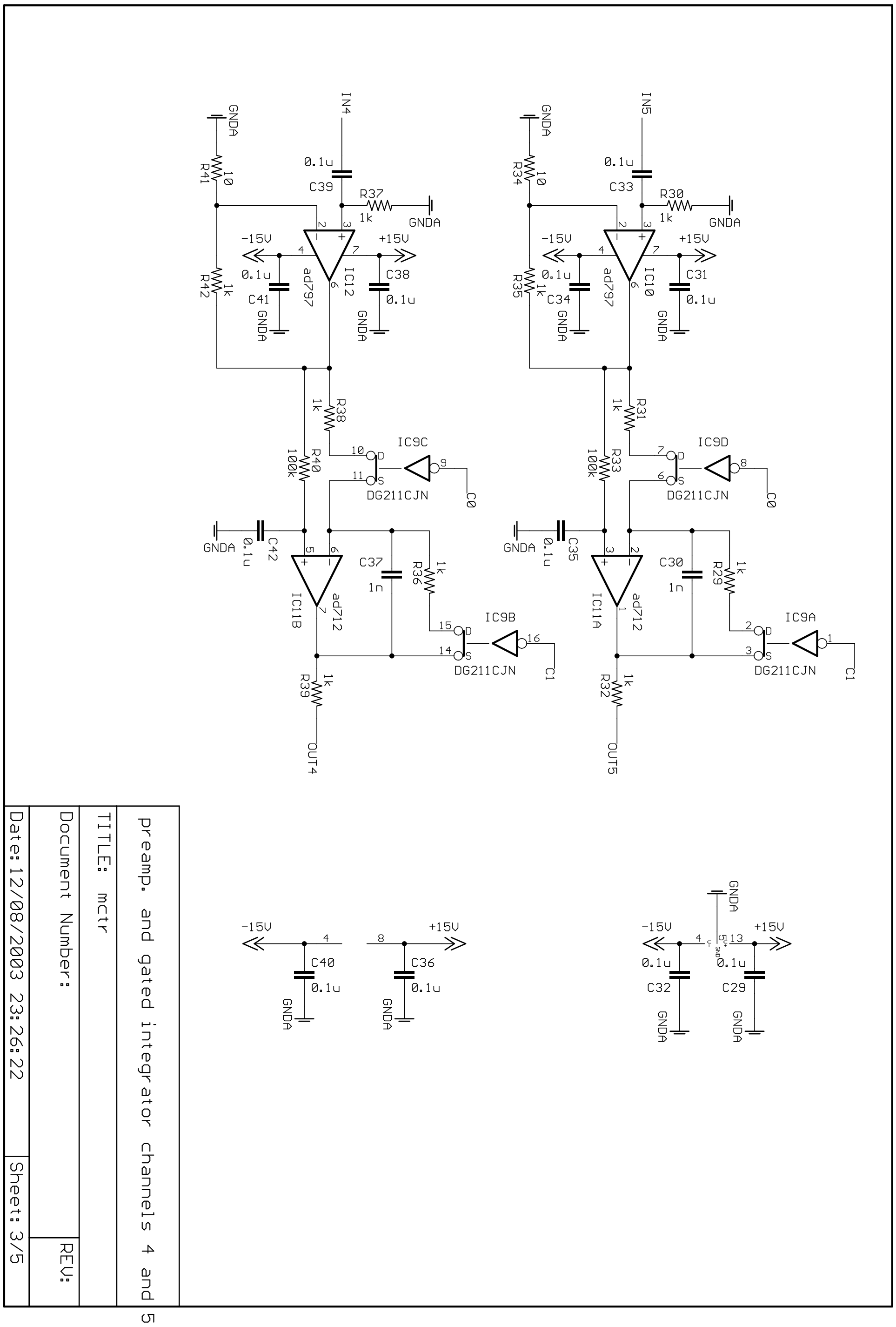

Figure 4: Detector channels 4 and 5.

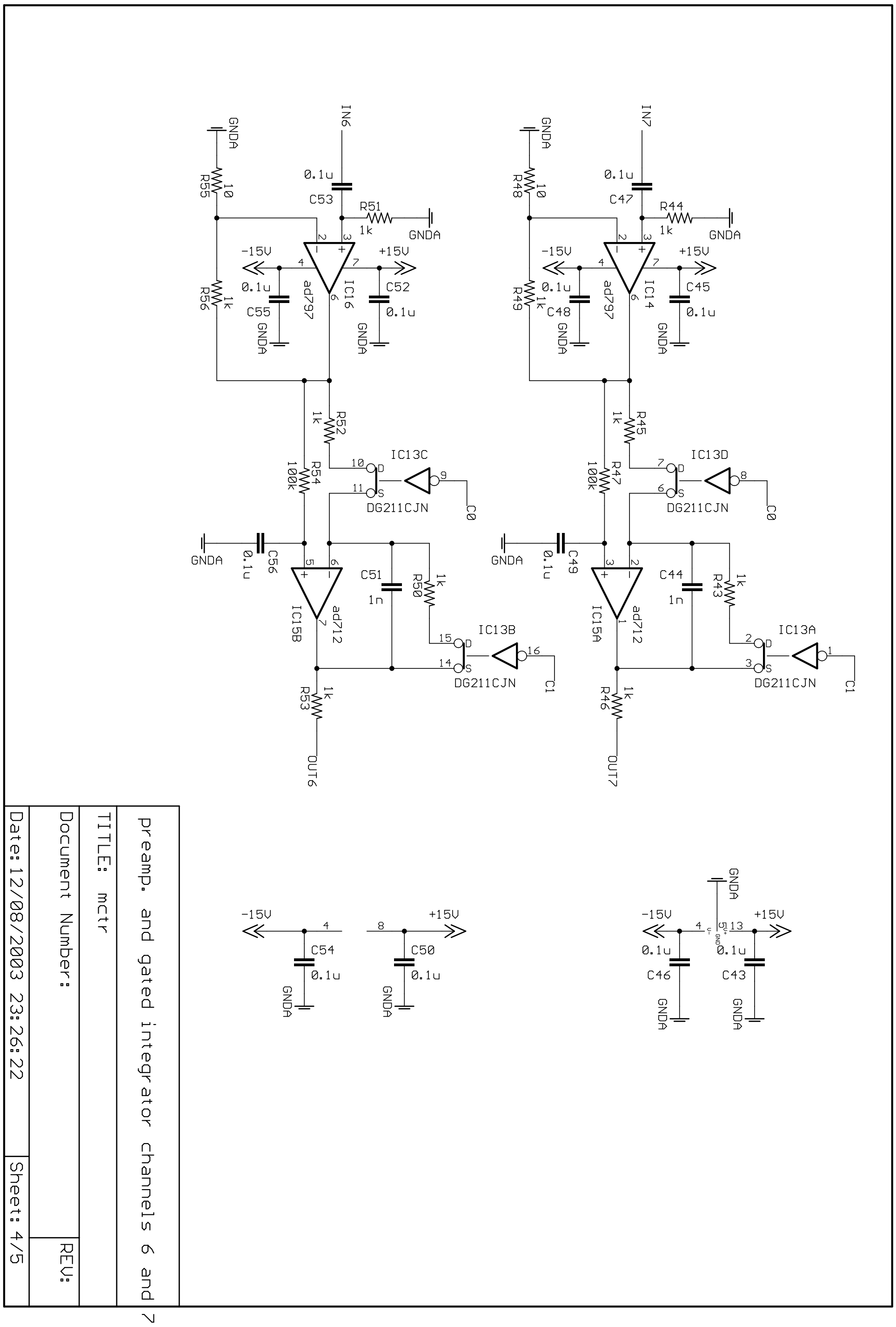

Figure 5: Detector channels 6 and 7.

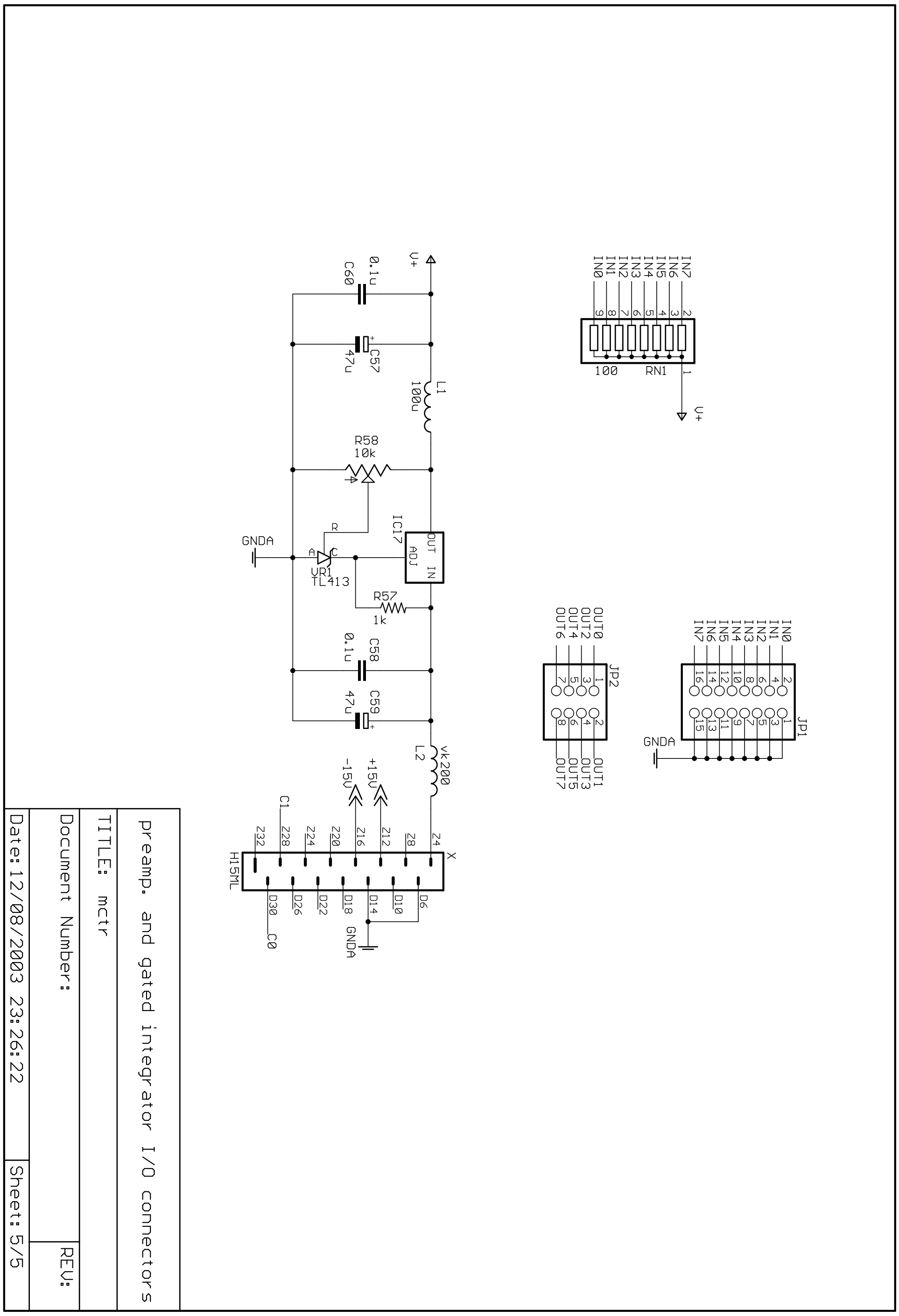

Figure 6: I/O Connectors.

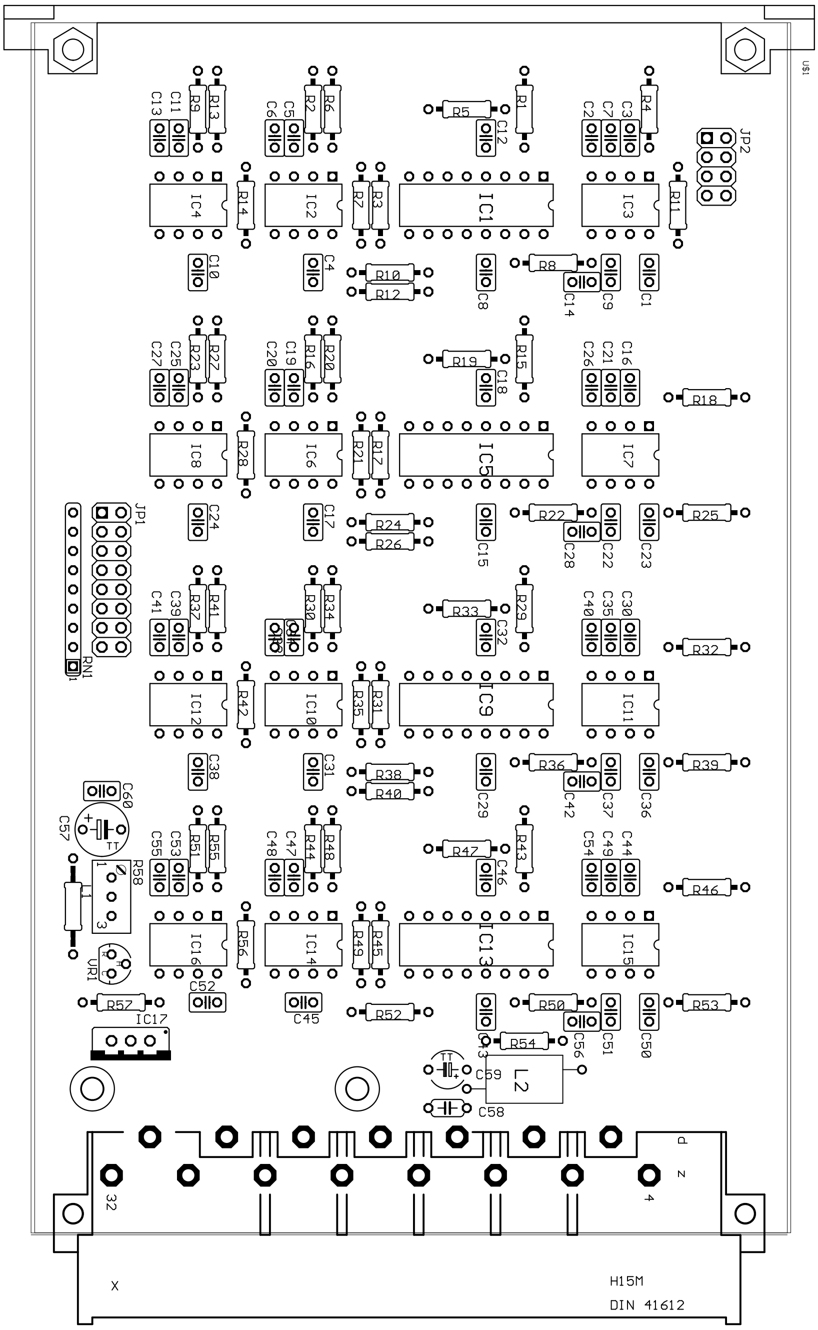

Figure 7: 8 Channels board silkscreen.

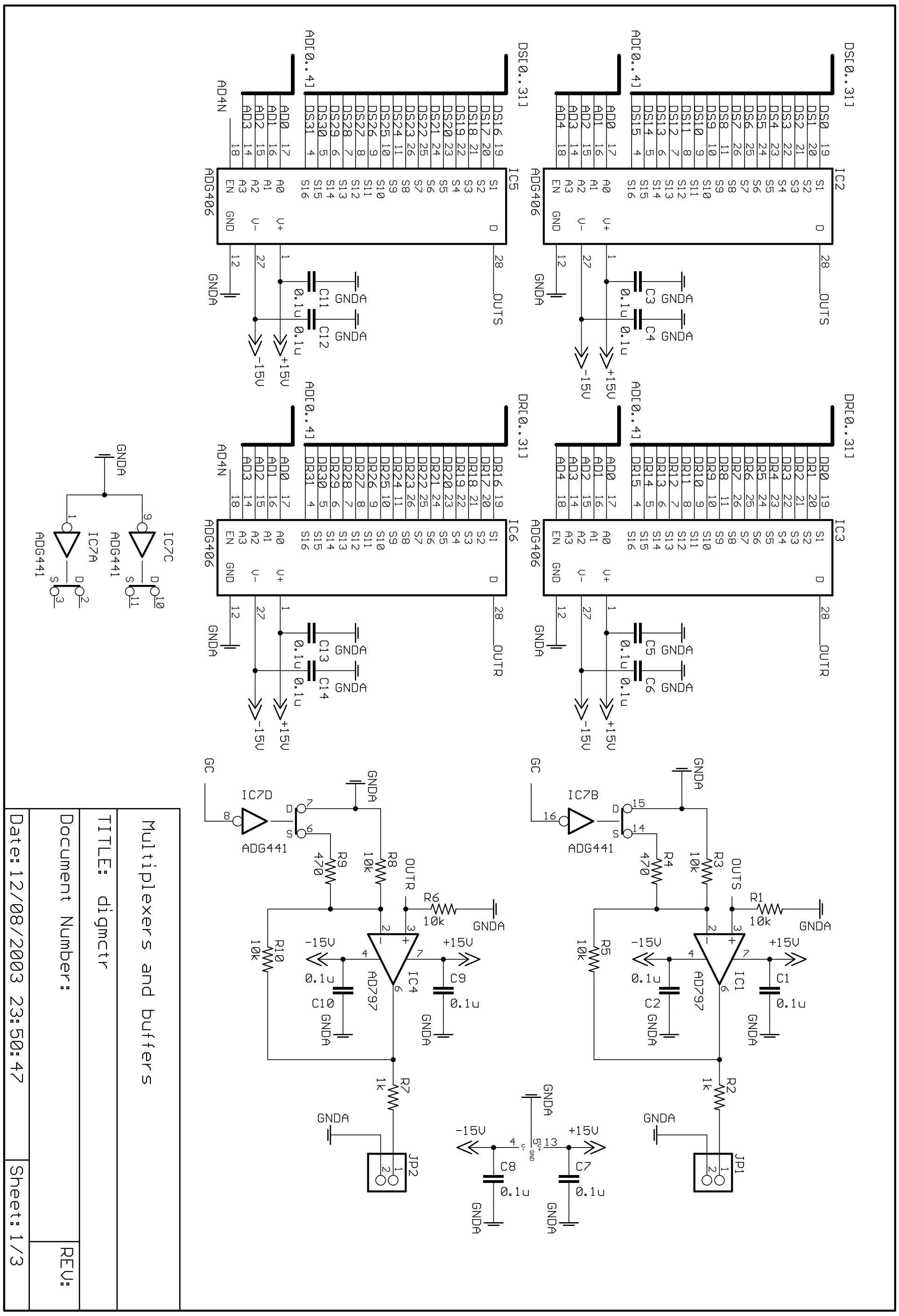

Figure 8: Output Multiplexers and buffers.

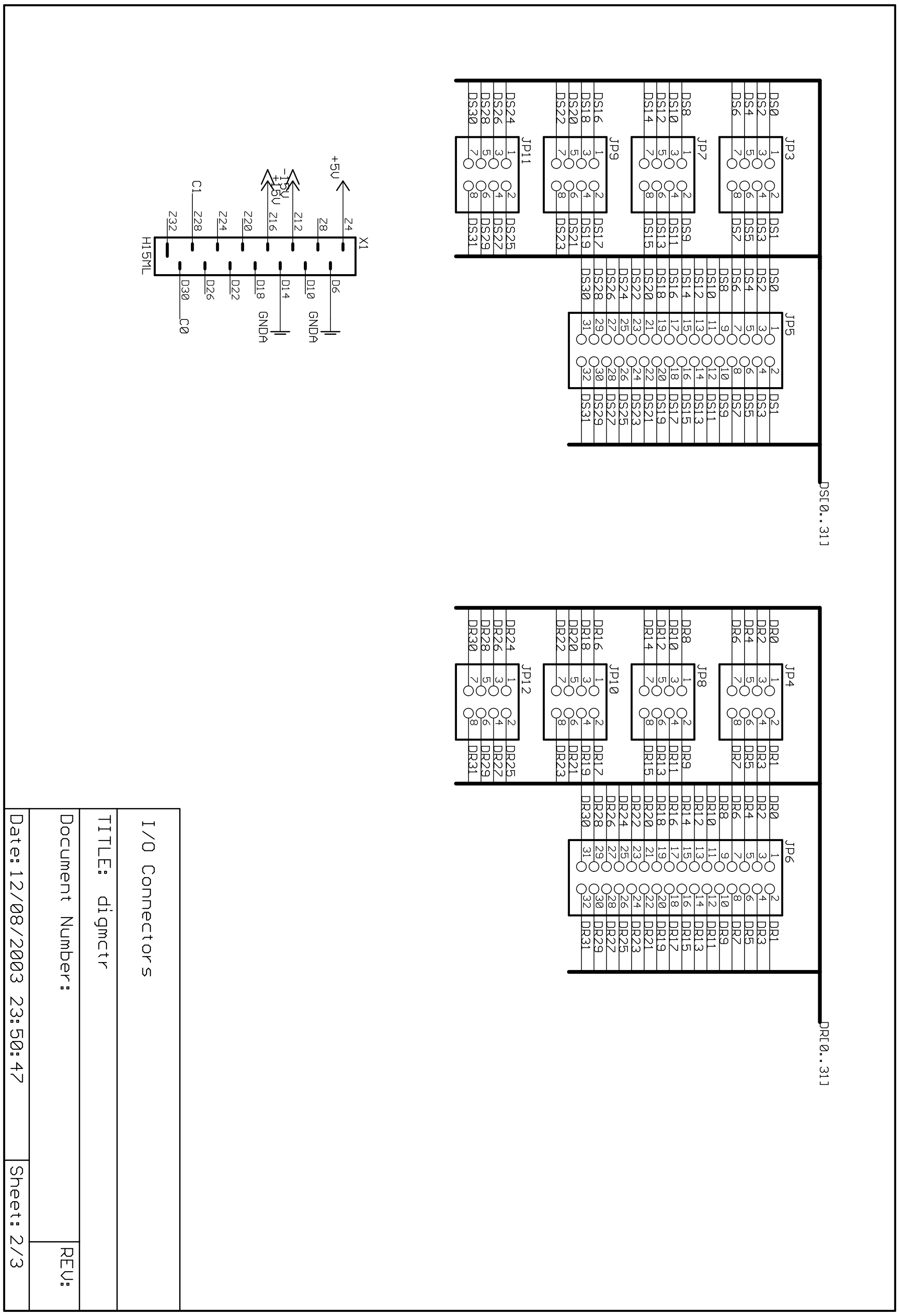

Figure 9: I/O Connectors.

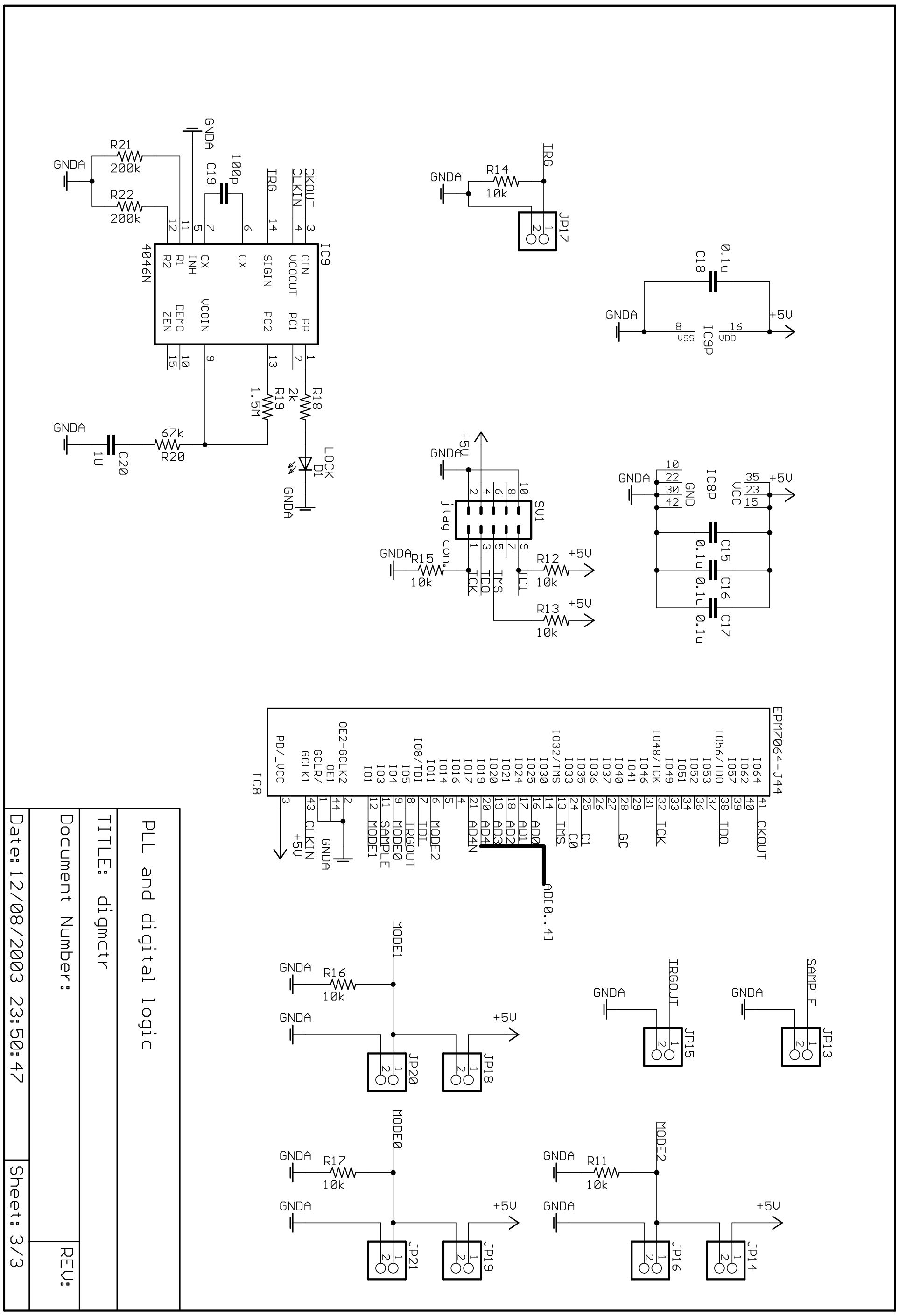

Figure 10: PLL and digital logic.

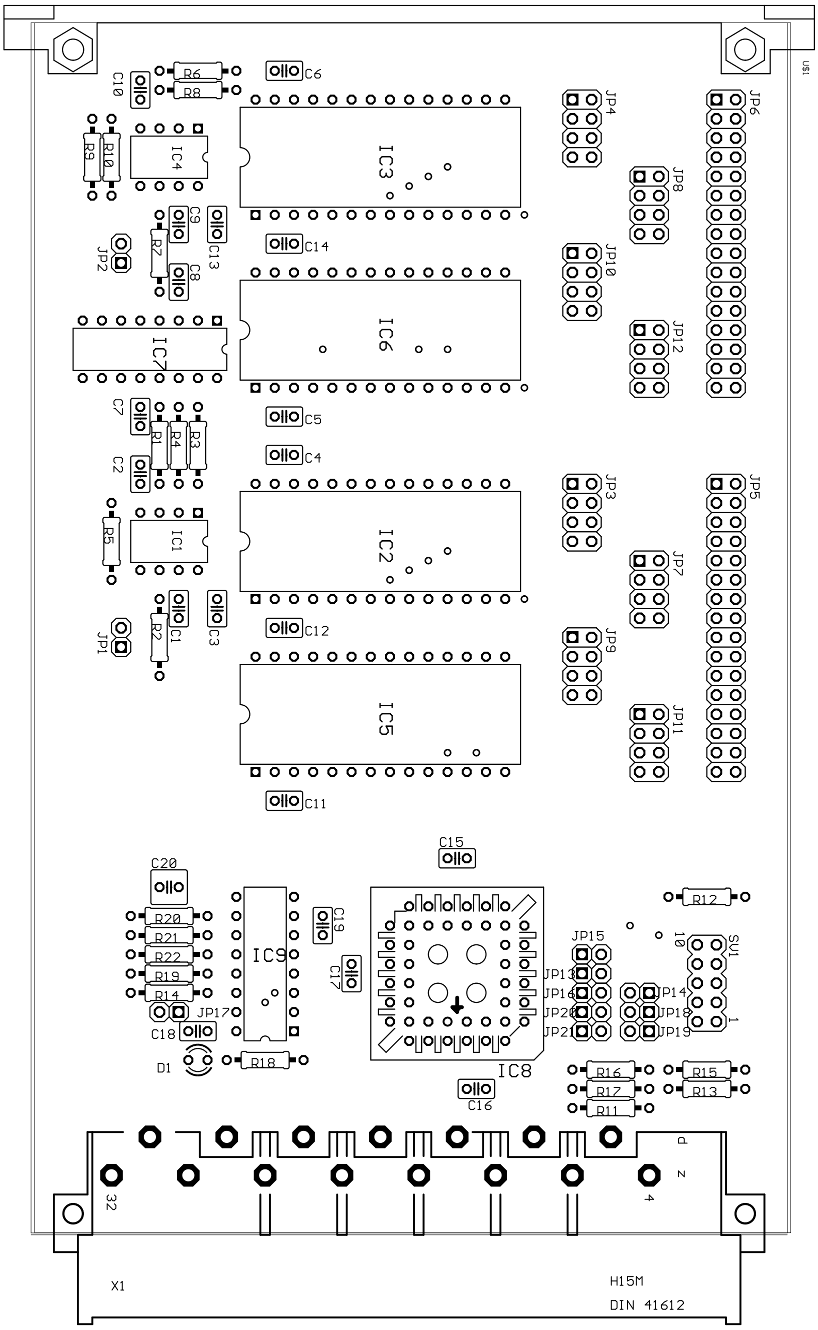

Figure 11: Control board silkscreen.

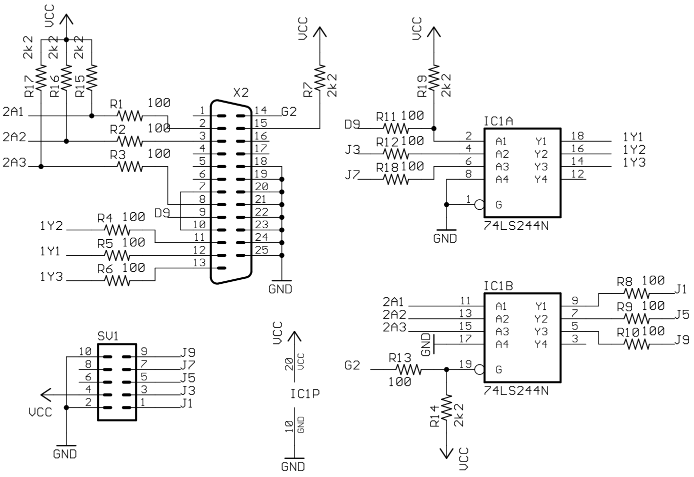

Figure 12: Schematics of the PC–JTAG interface.

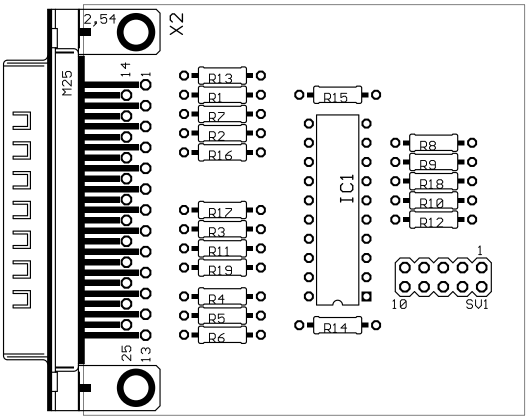

Figure 13: Top view of the PC–JTAG interface.

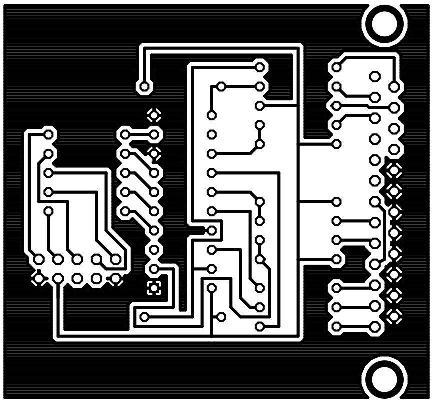

Figure 14: Solder side in scale 1:1 of the PC-JTAG interface.