Gliwice, 07-Dec-2003

Silesian University of Technology,

Faculty of Automatic Control, Electronics and Computer Science,

Institute of Computer Science,

Division of Microinformatics and Theory of Digital Automata.

ISA-AD

an 8-bit analog to digital converter card

project documentation

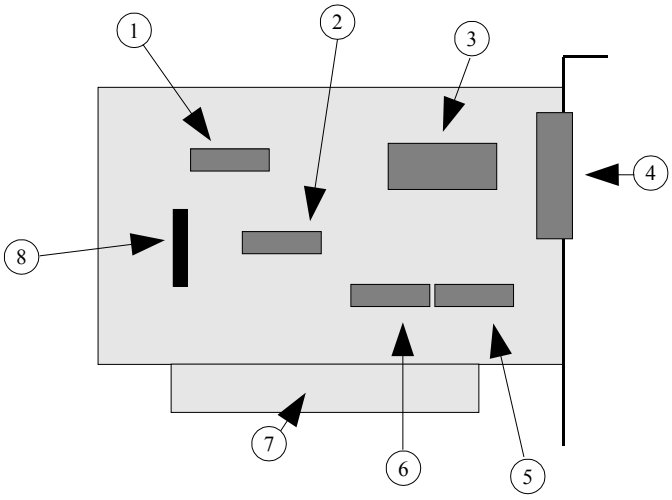
Tomasz Koprowski,

Makrokierunek, sem. 7

I. Overview

The ISA-AD is an 8-bit ISA card for PC class computers equipped with an appropriate slot. The card features eight analog channels multiplexed into a single analog-to-digital converter of 8-bit resolution. The external connector is a typical DB-25F. It works as eight channel analog input, a three channel logical (binary) output and a power supply connector for external "fan-out" box. Over-voltage protection should be implemented in the "fan-out" box – the card doesn't provide such a feature.

Figure 1:



1. control logic
2. address decoder
3. AD converter
4. ext. connector
5. input register
6. output register
7. ISA connector
8. address selector

II. Configuration

The base address of the card may be freely set in the range 0x000 through 0x03FF by placing jumpers on appropriate pins of the address selector "8". The most significant bit (MSB) of the address is at the top of the selector (when looking at the card as in Fig.1), the least significant bit (LSB) at the bottom. The selector uses "inverted logic" i.e. the presence of a jumper on a pair of pins denotes a logical "0", while its absence a "1".

Example 1: A typical configuration of ISA-AD is the address 0x0300 which written in binary is 1100000000b so the two topmost pins of the selector should be left open, while all the rest should be shorted with jumpers.

Note: The card doesn't support any auto-detection mechanism so it is crucial that the software is configured with the same address as the card. More on that subject in "Software Installation" section.

III. External connector pin-out

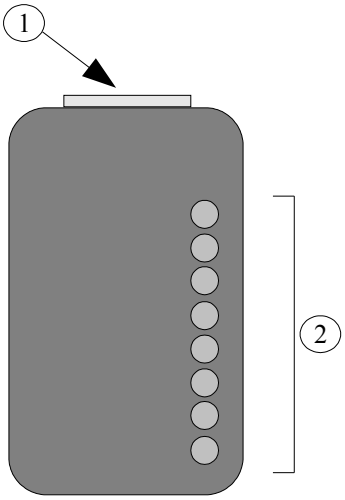
Figure 2:

Table 1:

1	analog channel 0 input
2	analog channel 1 input
3	analog channel 2 input
4	analog channel 3 input
5	analog channel 4 input
6	analog channel 5 input
7	analog channel 6 input
8	analog channel 7 input
9	not connected
10	logic output bit A
11	logic output bit B
12	logic output bit C
13	VCC (+5V)
14	GND
15	GND
16	GND
17	GND
18	GND
19	GND
20	GND
21	+12 V
22	not connected
23	not connected
24	not connected
25	not connected

IV. Fan-out box

Figure 3:



1. card connector
2. channels 0...7

The fan-out box connects to the ISA-AD card through a 1-to-1 25-wire ribbon cable connected to “1”. Input channels “2” are industrial-standard BNC connectors.

Table 2:

channel	features	bit
0	relay controlled voltage divider, over-voltage protection	A
1	relay controlled series capacitor (AC mode), over-voltage protection	B
2	relay controlled input disconnection, over-voltage protection	C
3	voltage divider, over-voltage protection	
4	over-voltage protection	
5	over-voltage protection	
6	over-voltage protection	
7	over-voltage protection	

The voltage dividers for channels 0, 2, 3 and DC-level for channel 1 are set with potentiometers inside the box (see “Fan-out schematics” for details). Input voltage for channels 2, 4, 5, 6, 7 should be in the range [0V, 5V], input voltage swing for channel 1 depends on potentiometer setting – usually <2.5V, input voltage for channels 0, 3 depends on respective potentiometers' settings.

V. Software installation

After unpacking the source tarball with:

```
tar -xzvf ISA-AD.tar.gz
```

but before compiling the software one must edit the `hw.h` file and set the `PORT` to the appropriate value set with jumpers on the card. In order to compile the software it is enough to:

```
cd ISA-AD
./compileall.sh
./linkrun.sh
```

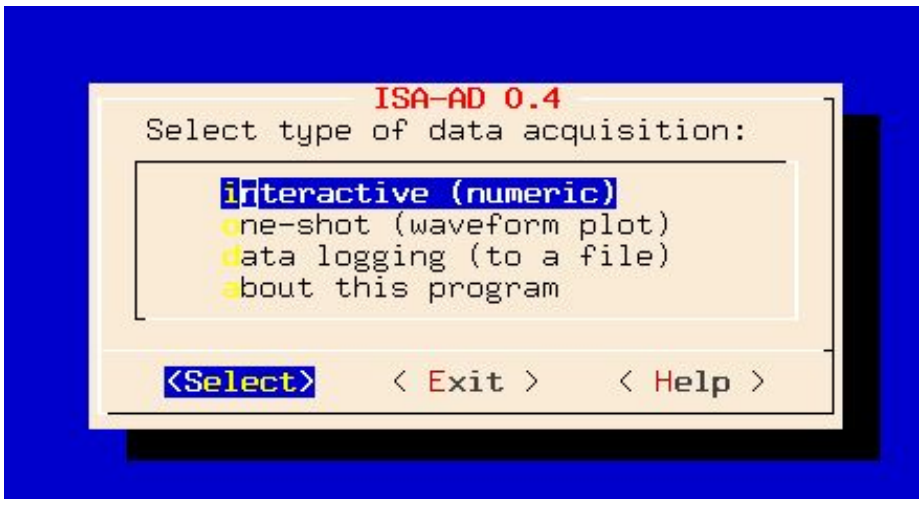
The binary called `isaad` will be created and launched, one can copy it later to any desired directory, keeping in mind that it has to be launched with super-user privileges (`uid0`).

The control software requires `ncurses` and `svgalib` libraries in order to compile and run.

VI. Software usage

When `isaad` is run it will present a splash screen, calibrate its internal delay-loop (more on that subject in "Conclusions") and present the main menu.

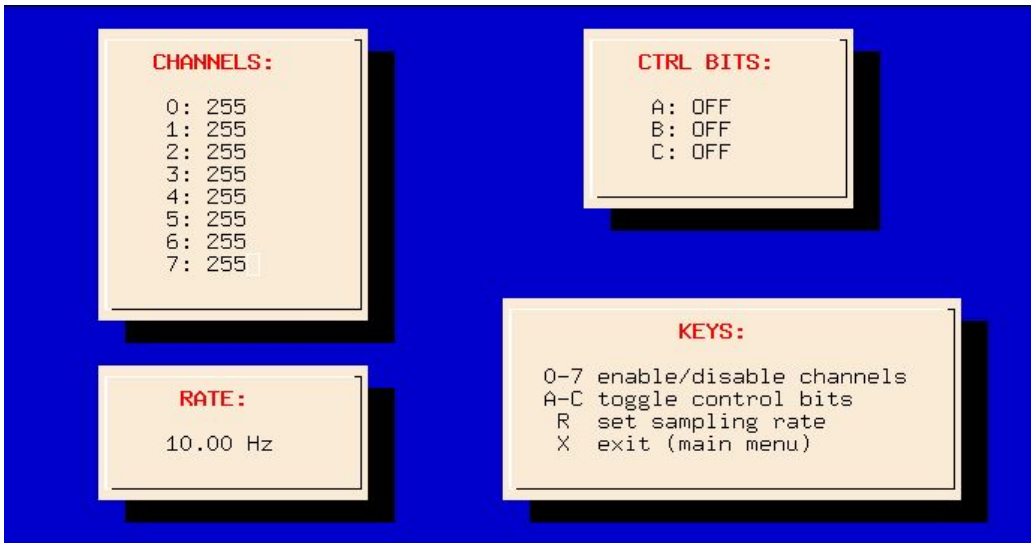
Figure 4:



Navigation through the menu is achieved with cursor keys and [Enter]. The Help screen gives information about available acquisition modes.

The following figure presents the “interactive” data acquisition mode.

Figure 5.



In this mode the user can watch the values of input voltages on selected channels. Channel selection toggled with keys [0] through [7], control bits (relays in fan-out box) are controlled with [A] - [C] keys, acquisition rate is set with the [R] key.

“One-shot” is a wizard that allows a user to acquire input waveform (and then present in graphically) after a series of a few simple questions (channels, sampling rate, etc.). The acquisition is started either when the user presses [Enter] (no trigger) or when a selected channel goes either above or below a given value.

“Data logging” works in a similar fashion as “One-shot” but with the difference that it was designed for long-term acquisitions with low sampling frequencies. It stores data into a user-selected CSV file that can be easily imported to a spreadsheet program for further analysis. The acquisition process may be interrupted by hitting any key, however the response delay may be up to a single inter-sample period long.

VII. Theory of operation

When the PC tries to access a device on ISA bus it sets the BALE line (B28 contact on the ISA slot) to a logic “0” state. This condition enables the comparison of the address present on lines A2 through A9 (contacts A29-A22) in the 74HCT688 8-bit magnitude comparator to the address set by eight MSBs of the address selector jumpers. When the two numbers match, the comparator pulls its /P=Q line to logic “0” which in turn enables further condition checks in GAL16V8.

The GAL16V8 was programmed for two operations working independently from each other: a partial address decoder (two LSBs of the address) with ISA bus signals interpretation and a frequency divider.

A condition when /P=Q is “0”, LSBs of the address (lines A0, A1 – contacts A31, A30) match with appropriate values of the address selector and IOW line (contact B13) is “0” is interpreted as a write to the card. In such a case ICL line (pin 18 of the GAL) is set to “1” which signals the input register 74HCT574 (IC4, pin 11) to latch data from data bus consisting of D0 through D7 (contacts A9-A2). The bits of the input byte are interpreted as follows:

0 (LSB)	analog multiplexer address bit A (LSB)
1	analog multiplexer address bit B
2	analog multiplexer address bit C (MSB)
3	analog multiplexer Address Latch Enable
4	start AD conversion
5	control bit (relay) A
6	control bit (relay) B
7 (MSB)	control bit (relay) C

Note: analog multiplexer and analog to digital converter are implemented in the same chip – the ADC0809.

When /P=Q is “0”, LSBs of the address (lines A0, A1 – contacts A31, A30) match with appropriate values of the address selector and IOR (contact B14) is “0” the GAL recognizes a read request to the card and sets line OOE (pin 19) to “0”. The OOE line is connected to OE of the 74HCT574 output register (pin 1) when it is “0” the outputs of the register become active (ie. leave the high impedance state) and its contents are placed on the data bus D0 through D7 (contacts A9-A2).

The second process the GAL implements is frequency division – the ISA bus clock of frequency about 8 MHz (depending on motherboard and sometimes BIOS configuration) present on CLK line (contact B20) is divided by 8. The divider is implemented as a simple 3-bit counter with outputs on pins 13, 14, 15. The 13th pin (CCLK=CLK/8) feeds the clock input (pin 10) of the ADC0809 converter, which can work with frequencies up to 1.28 MHz.

In order to begin AD conversion the control program must issue a series of writes to the card: first, the channel number must be written, then a proper analog multiplexer Address Latch Enable (AMALE) signal must be generated (channel number ORed with 0x08), AMALE must be deactivated (write of channel number) and a converter start signal must be activated (write of 0x10).

The conversion time depends on ISA bus frequency, for 8 MHz it should be about 78 μs, the control program should wait at least that long before issuing a read to the card. When the conversion time elapses, a single read to the card gives a result of the last conversion.

Note: for the relays in fan-out box to work reliably each write to the card should be ORed with the bits representing respective relays,

Example 2: In order to latch channel number 3 to the analog multiplexer without disturbing the operation of turned-on relays A and C one should write 10101011b to the card.

The analog inputs of the ADC0809 are connected directly to the external connector. Its data output bus (pins 17, 14, 15, 8, 18, 19, 20, 21) is connected to the inputs of output register (IC1) 74HCT574. When ADC0809 finishes a conversion, it sets its EOC line (pin 7) to "1" which makes the IC1 latch the value. The Output Enable (pin 9) of the converter is tied to VCC.

The lower reference voltage for ADC0809 (pin 16) is tied to ground while the upper one (pin 12) is fed from an integrated circuit +5V voltage source, the LM336-5V.

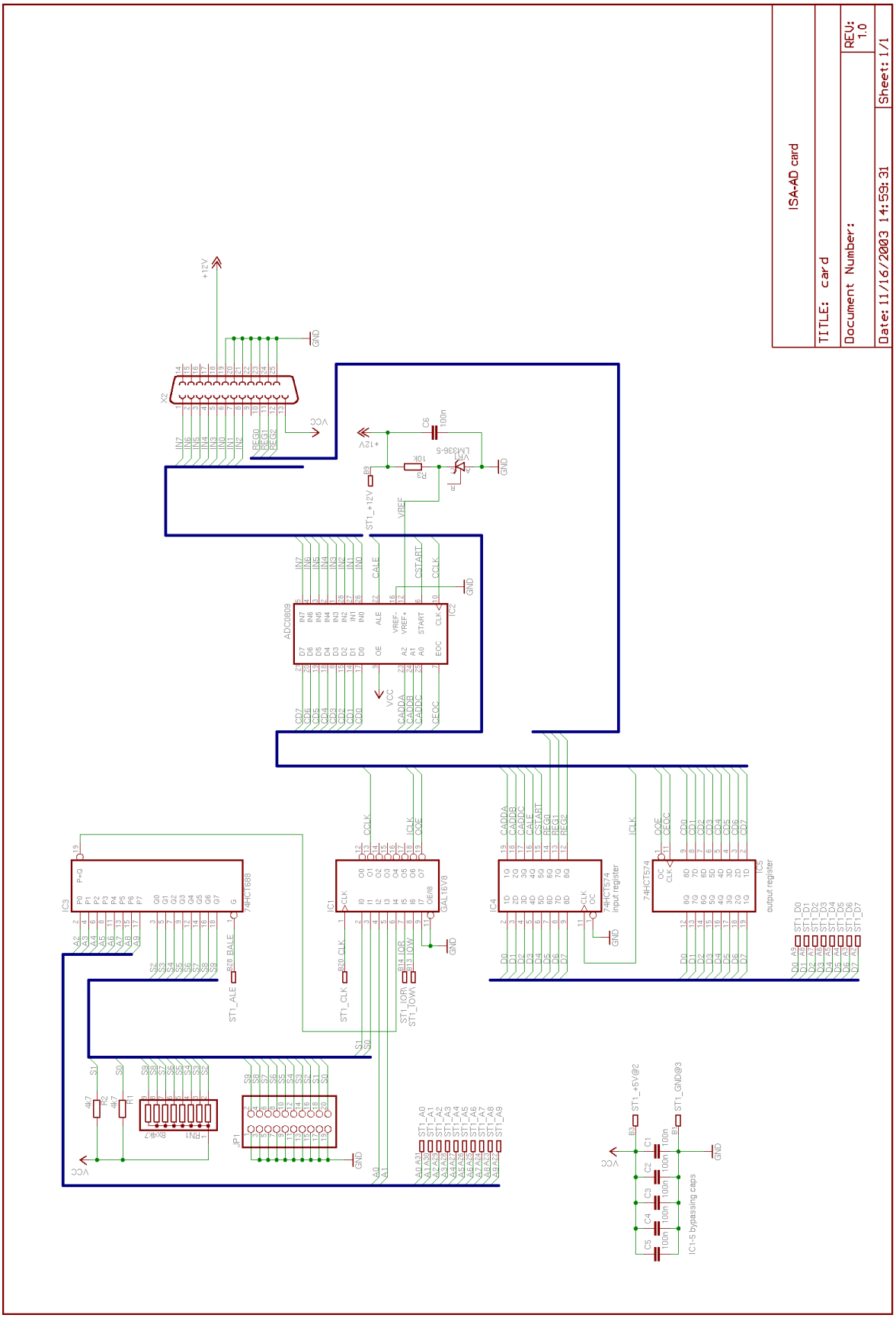
In order to reduce supply voltage noise 100nF capacitors (one per IC) are placed on the board as close to the chips as possible.

VIII. The GAL structure

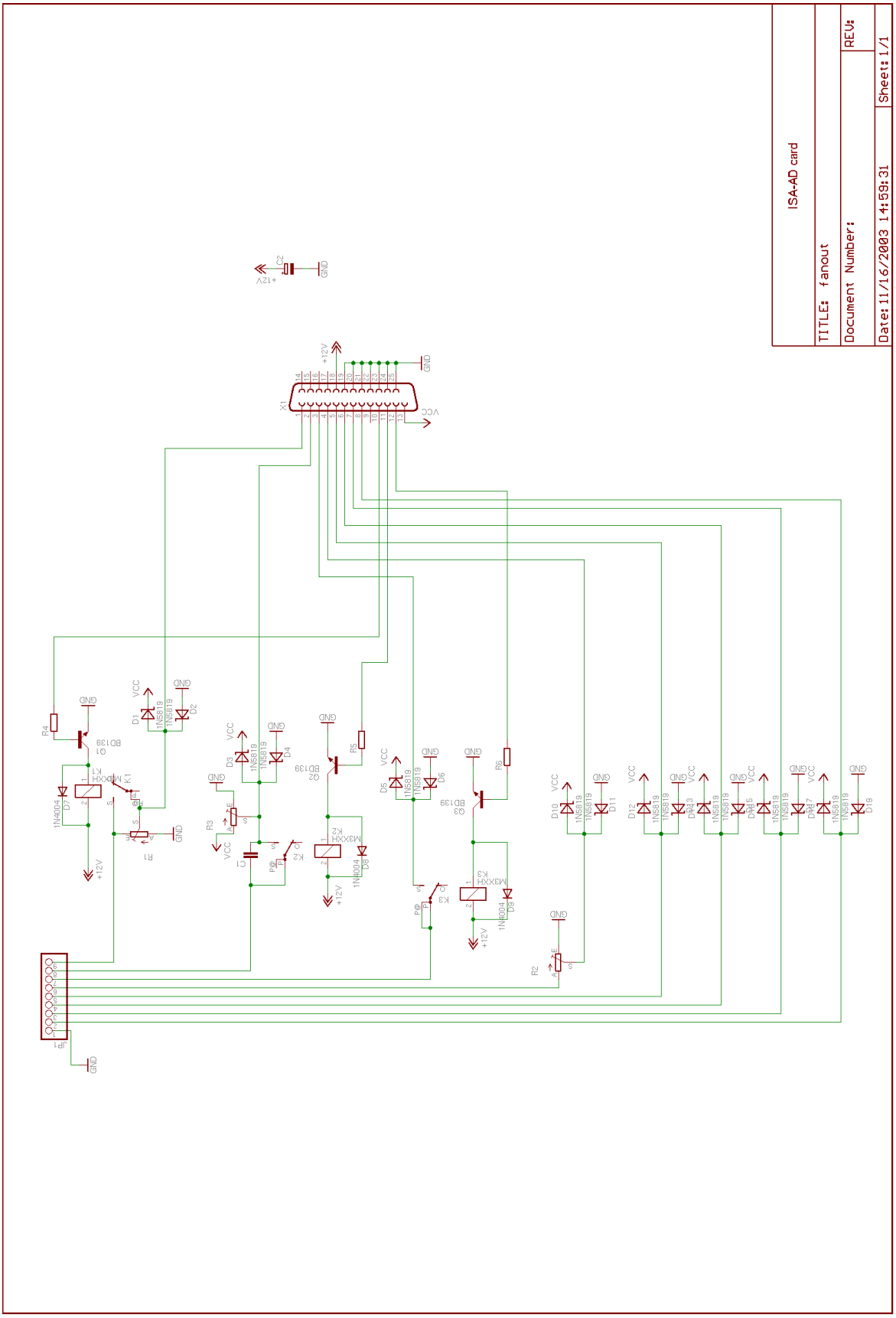
The logic structure of the GAL was programmed in ABEL hardware description language:

```
module ISAlogic
title 'GAL based clock divider and ISA decoder'

ISAlogic device 'P16V8';
      CLK pin 1;
      Q2,Q1,Q0 PIN 13,14,15 ISTYPE 'REG';
      DIVOUT=[Q2,Q1,Q0];
      S1,S0 PIN 2,3;
      A0,A1 PIN 4,5;
      IOR PIN 7;
      IOW PIN 8;
      PQ PIN 6;
      OOE PIN 19 ISTYPE 'COM';
      ICL PIN 18 ISTYPE 'COM';

equations
      DIVOUT.clk=CLK;
      DIVOUT:=DIVOUT.fb+1;
      ICL = !PQ & (S1==A1) & (S0==A0) & !IOW;
      OOE = !(!PQ & (S1==A1) & (S0==A0) & !IOR);

END ISAlogic
```

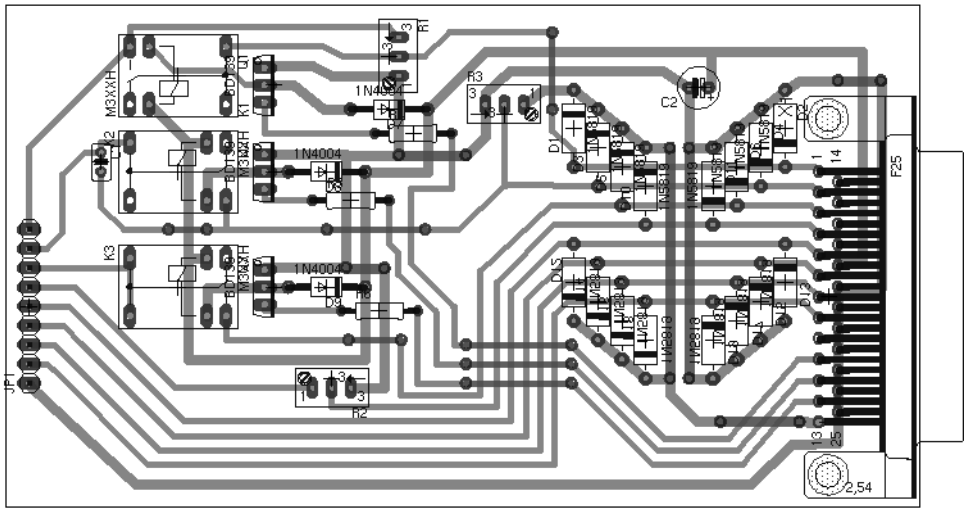
IX. Card schematics

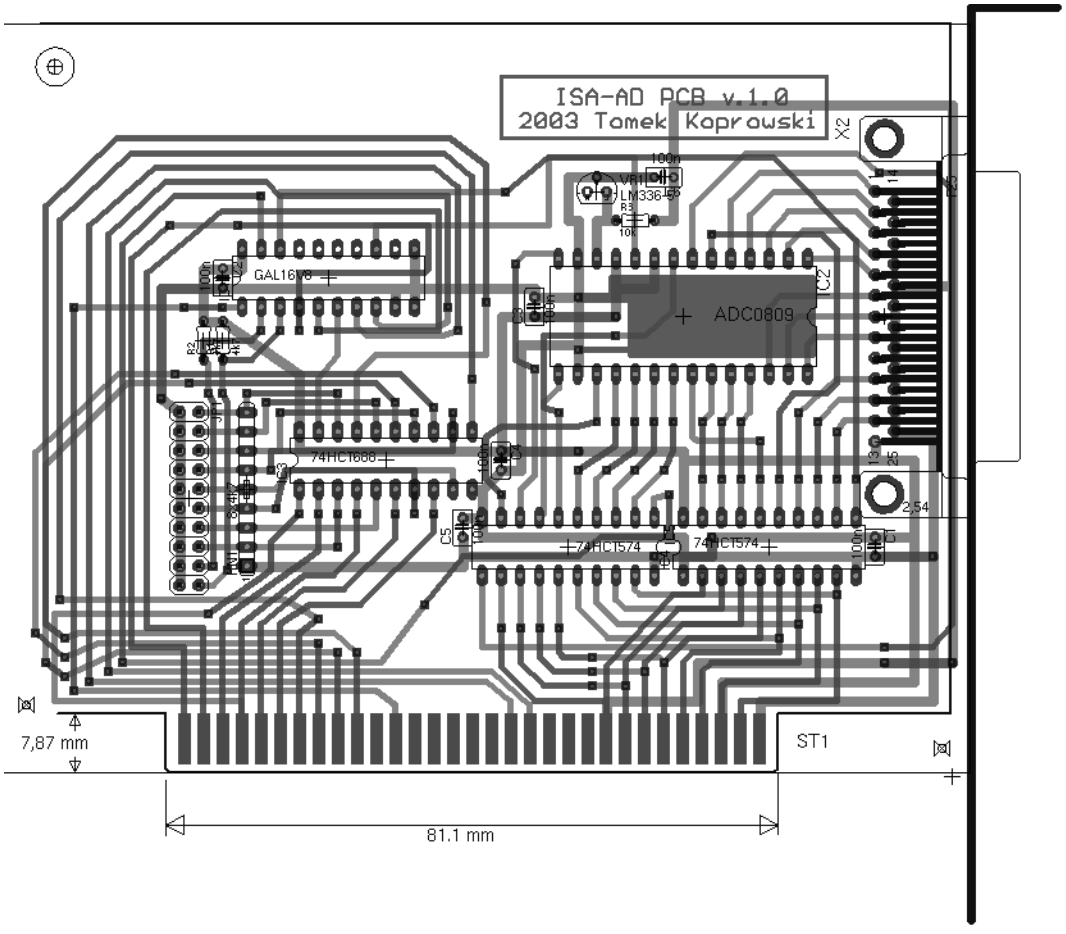


X. **Fan-out schematics**



XI. Board layouts





XII. Conclusions

During the development of the ISA-AD card some problems were solved in sub-optimal ways, which was unfortunately realized after the project was ready.

First of all it would be much better to place the AD converter chip in the fan-out box – this way crosstalk between channels and external noise could be reduced. For further reduction of noise the fan-out box case should be made of EM shielding materials (aluminum, steel or copper plates).

The main source of crosstalk between channels is the flat ribbon cable connecting fan-out box to the card, so if moving the ADC outside the card would be for some reasons undesired, it would make sense to separate channel wires by ground wires or change the cable for a multi-shielded one.

In order to save some board space it would be desirable to use surface-mount versions of the 74xxx ICs. During the assembly of the board it was realized that some of the tracks (mainly interconnections of the registers to ADC and bus to comparator) could be moved from bottom layer to top layer thus reducing the number of vias and in turn the assembling time. The board could easily get away with 26 vias less, without much redesign of the layout, at the cost of soldering through-hole DIP packages from the component side, which isn't a great problem when assembling by hand using a fine-tip soldering iron.

The board layout could be further improved by using a registered version of the Eagle schematic/PCB design program. A freeware version was used instead, which had a limitation of the maximum board space occupied by elements, the space occupied by tracks fortunately wasn't limited.

The second class of problems was introduced by choosing Linux as the host system for the program. This system lacks high-precision timers and delay functions that were needed for proper timing of the acquisition (especially in the "one-shot" mode). It was solved by implementing busy-wait loops calibrated at program start. The program requests form the system to be scheduled on "as close to real-time as possible" basis (`SCHED_FIFO` scheduler with a priority of 99), however Linux is not an RTOS – only user-space processes are preemptive, kernel is not. It was first assumed that the software will be implemented on RTLinux (a real-time kernel running Linux kernel as one of its processes, more on http://www.rtlinux.org/) but the idea was abandoned due to the lack of documentation for the free RTLinux, which has forked to a commercial product.

It was considered to implement a selection of I/O port in the program instead of setting it at compile-time, however this idea was discarded due to security reasons (when the binary would be installed SUID root, a malicious user could possibly destroy contents of disks by a specially crafted sequence of I/O port selections and conversion requests).

The most annoying source of problems at the design stage was contradiction between different sources of information regarding the ISA bus operation, mainly mutual timing of signals, but a working solution was finally found.