Client Server Architecture based Embedded Data Acquisition System on PC104

Jigneshkumar J. Patel, Rachana Rajpal, Praveena Kumari, P.K. Chattopadhyay and R. Jha*

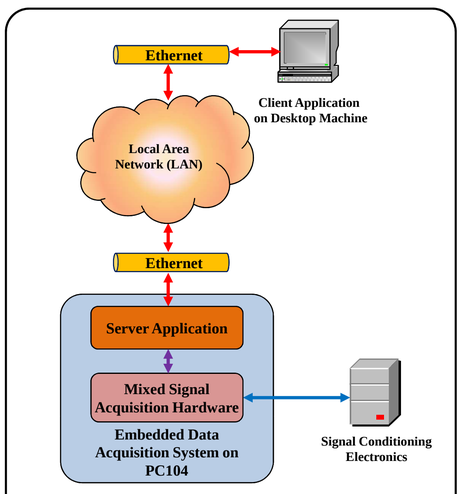
Institute for Plasma Research, Gandhinagar, Gujarat, India

**jjp@ipr.res.in*

INTRODUCTION

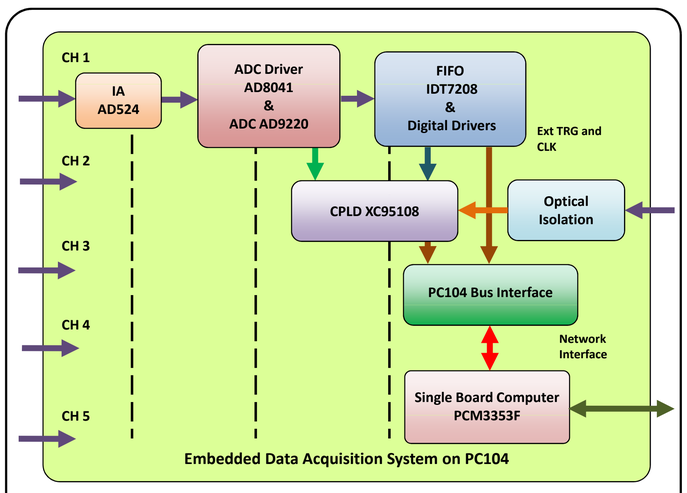
The data acquisition system is designed on embedded PC104 platform Single Board Computer (SBC) with running Windows XP Embedded operating system. This is a multi channel system which consists of 12 Bit, 10 MSPS Analog to Digital Converters with on board FIFO memory for each channel. The digital control and PC104 bus interface logic are implemented using Very High Speed Hardware Description Language (VHDL) on Complex Programmable Logic Device (CPLD).
The system has provision of software, manual as well as isolated remote trigger option. The Client Server based application is developed using National Instrument CVI for remote continuous and single shot data acquisition for basic plasma physics experiments. The software application has features of remote settings of sampling rate, selection of operation mode, data analysis using plot and zoom features. The embedded hardware platform can be configured to be used in different way according to the physics experiment requirement by different top level software architecture. The system is tested for different physics experiments. The detailed hardware and software design, development and testing results are discussed in the paper.

System Diagram



Basic System Block Diagram

Block Diagram



Block Diagram of Embedded DAQ System (5 Channels)

Basic Hardware Features

- Embedded hardware platform with on board analog, digital, processing and communication blocks.
- Consists five channels of high speed pipeline ADC channels with precision instrumentation amplifier and high speed ADC driver.
- Precision Instrumentation Amplifier AD524 with 25MHz GBP.
- 160 MHz, Rail to Rail Amplifier AD8041 as a ADC Driver.
- Single ended and differential input signal option.
- 12 Bit, 10 MSPS pipeline ADC AD9220AR.
- On board 64 KB FIFO (First In First Out) memory for each channel
- System timing, control logic and PC104 bus interface are implemented in CPLD using VHDL.
- CPLD interfaced with on board single board computer (SBC) by PC104 bus.
- Two mode of operations (1) Single Shot (2) Continuous
- SBC runs on Windows XP Embedded operating system installed on CF (Compact Flash) card with Ethernet and USB Connectivity.
- External optically isolated trigger and clock option.
- Designed and assembled in industry standard 6U, 16T enclosure.

VHDL Program Logic

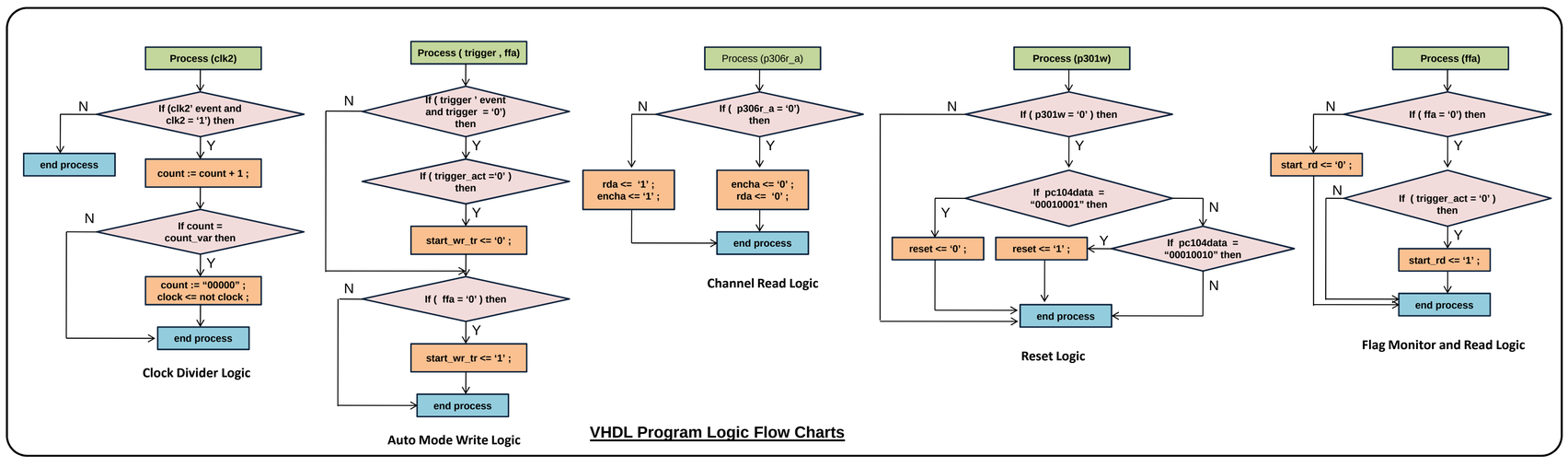
All embedded control logic of hardware is implemented in CPLD using VHDL. The main implemented components are as mentioned below

- PC104 address and data decoder.
- ADC clock logic and FIFO write and read logic.
- Clock divider and sampling rate selection logic.
- Mode selection and flag status monitoring logic.

```
--For clock divider
process(clk2)
variable count : std_logic_vector (6 downto 0)
:= "0000000";
begin
  if(clk2'event and clk2 = '1')then
    count := count + 1;
    if count = count_var then
       count := "0000000";
       clock <= not clock;
    end if;
  end if;
end process;
```

```
--To start and stop the ADC clock in
trigger mode
process(trigger,ffa)
begin
if (trigger 'event and trigger = '0')then
    if(trigger_act = '0')then
       start_wr_tr <= '0';
    end if;
end if;
if (ffa = '0')then
    start_wr_tr <= '1';
end if;
end process;
```

VHDL Code



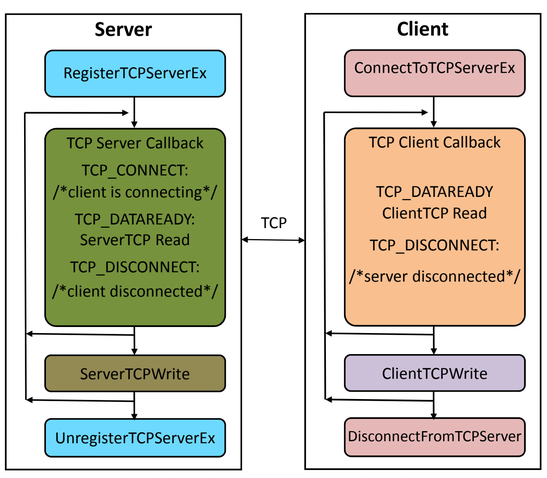
VHDL Program Logic Flow Charts

PC104 Address decoding

PC104 Address	Decoding
0x306,0x308,0x30A, 0x30C, 0x30E	16 bit read cycle for each channel data read out (Read)
0x301	FIFO Reset (write)
0x302	FIFO Clock Enable (write)
0x303	Trigger Status (Read)
0x304	Trigger Mode (write)
0x305	Clock variable (write)
0x309	Half Flag Status (Read)

Software Architecture

- The client server architecture is the most suitable for this type of application, as it requires minimum installations at client side and uses local area network for data exchange.
- The software development is done in National Instrument's LabWindows/CVI, which is ANSI C based development environment on windows platform which also provides good support for graphical user interface (GUI) development.
- The LabWindows/CVI TCP Support Library provides easy-to-use callback functions to create TCP server and client applications.
- The Callback functions provide the mechanism for receiving notification of connection initiation, connection termination, and data availability.



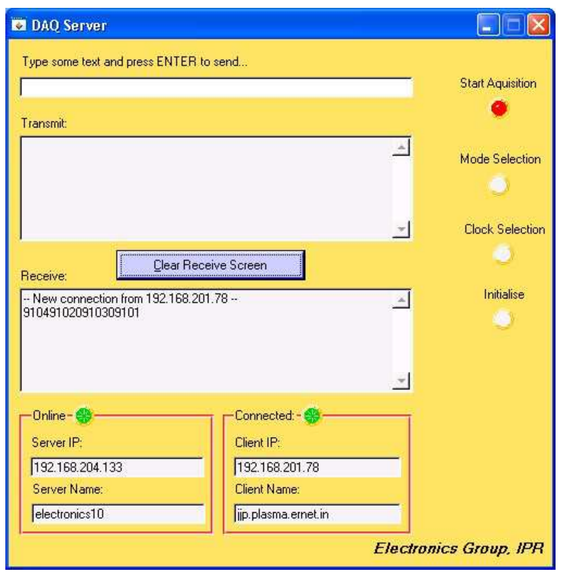
LabWindows/CVI TCP Support functions

Software Development

The software development is divided in main two parts server application and client application. The custom protocol is designed using the data packets, which are exchanged between server and client application to establish the operation, control and status monitoring of the embedded hardware.

Server Application

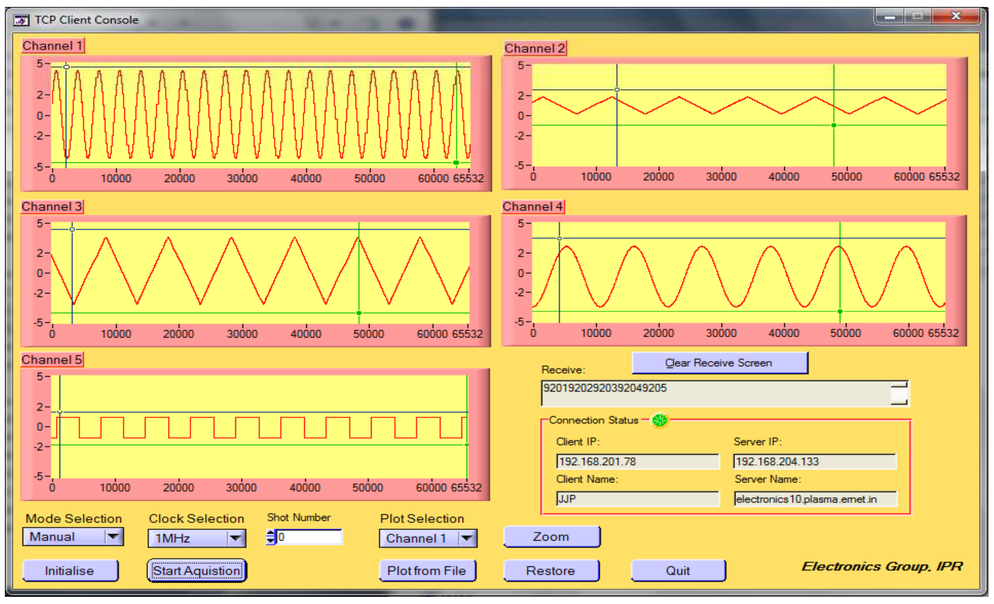
- The Server application runs on the embedded hardware platform and it mainly controls the hardware parameters of the module.
- Server application generates the control commands and reads the status of the hardware through PC104 bus.
- The connections details and the commands received from the client application are displayed in the text boxes in the server application GUI.



GUI of Server Application

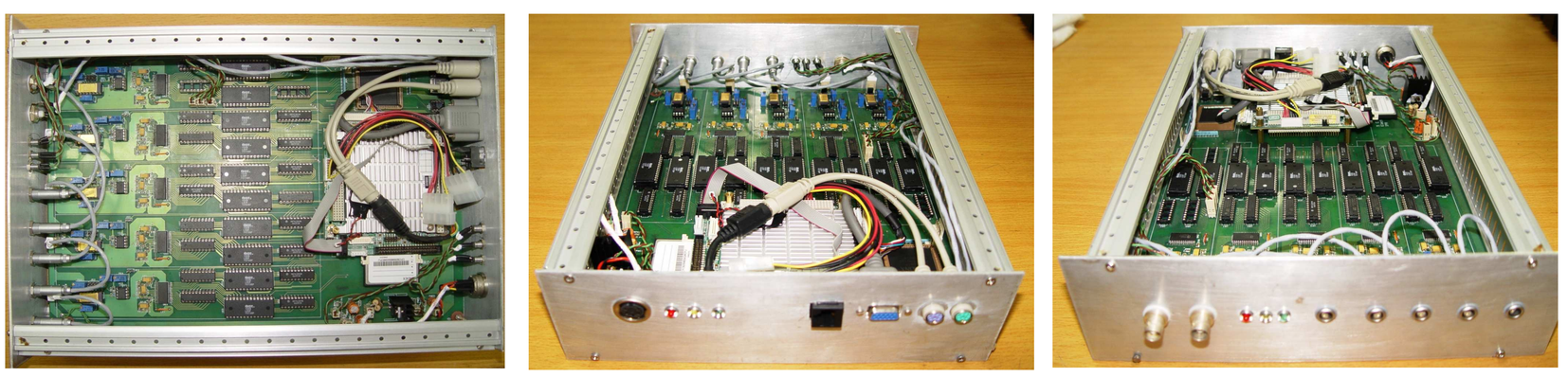
Client Application

- The client application runs on any networked computer and it remotely controls the embedded data acquisition system.
- It supports plotting of all channels and some basic data analysis utilities like zoom, restore and plotting from files.
- The selected data acquisition parameters are embedded in the data packets with the starting string number '91' followed by the data parameter, which are sent to server application.
- After completion of the data acquisition, the client application receives the data of each channel which is stored in the file based database for particular defined shot number.
- The TCP connection details, the proper file reception and the acquired data are shown in GUI.



GUI of Client Application

Embedded Hardware System



Embedded DAQ System